

# 2014

Facultat d'Informàtica de  
Barcelona (UPC)

Joan Viladrosa Riera

## [ MEMÒRIA PFC: PLATAFORMA DE MUSEUS VIRTUALS 3D ]

**Titulació:** Enginyeria en Informàtica (2003)  
**Tribunal:** *President:* Guillermo Godoy Balil  
*Vocal:* José Luis Ruiz Muñoz  
*Secretari:* Borja Vallés Fuente  
*Codirectors:* Alejandro Ballarin Latre  
Borja Vallés Fuente

## Índex

1.	Motivacions .....	3
2.	Introducció .....	4
2.1	El museu tradicional.....	4
2.2	Museus Virtuals a l'actualitat .....	4
2.3	Museus Virtuals en un futur.....	6
2.4	El mercat de la plataforma .....	7
3.	Proposta .....	8
3.1	Requisits funcionals .....	8
3.2	Requisits no funcionals .....	9
3.3	Arquitectura del sistema .....	9
4.	Planificació.....	11
4.1	Històries d'usuari .....	11
4.2	Disseny tècnic .....	13
4.2.1.	Llenguatge de programació .....	13
4.2.2.	Framework web .....	13
4.2.3.	Entorn de treball web .....	14
4.2.4.	Patró Model-Vista-Controlador .....	15
4.2.5.	Patró Data Mapper .....	16
4.2.6.	Mapatge d'objectes relacional.....	16
4.2.7.	Internacionalització .....	16
4.2.8.	HTML5 + CSS3 + Bootstrap + jQuery .....	17
4.2.9.	Motor Visualitzador 3D.....	18
4.2.10.	Visual Studio C# Express .....	19
4.3	Metodologia de Treball .....	20
4.3.1.	Control de versions de codi .....	20
4.3.2.	Servidor de producció.....	20
4.3.3.	Metodologies àgils de desenvolupament.....	21
4.4	Pressupost Inicial .....	23
5.	Desenvolupament.....	25
5.1	Sprint #1: Set-up inicial i gestió usuaris .....	25
5.2	Sprint #2: Gestió museus i personal .....	28

5.3	Sprint #3: Gestió obres.....	31
5.4	Sprint #4: Editor mapes: sales .....	33
5.5	Sprint #5: Editor mapes: parets.....	37
5.6	Sprint #6: Webservice iMuseums.....	39
5.7	Sprint #7: Visualitzador 3D .....	42
5.8	Sprint #8: Sistema d'estadístiques.....	45
5.9	Sprint #9: Documentació i Memòria.....	47
5.10	Diagrama de taules Base de Dades.....	48
6.	Conclusions.....	49
6.1	Cost final i càlcul d'hores.....	49
6.2	Conclusions sobre el projecte .....	50
6.3	Conclusions personals.....	50
6.4	Treball futur.....	51
7.	Glossari.....	54
8.	Bibliografia.....	56
9.	Annexos.....	58
9.1	API pública webservice .....	58
9.1.1.	Mòdul museus.....	58
9.1.2.	Mòdul rols.....	59
9.1.3.	Mòdul obres d'art.....	60
9.1.4.	Mòdul dissenyador .....	61
9.1.5.	Mòdul iMuseums.....	62
9.1.6.	Mòdul iMuseums3D .....	62
9.1.7.	Mòdul estadístiques .....	63
9.2	Format JSON iMuseums3D.....	64
9.3	Cas d'ús: Crear sala .....	65

## 1. Motivacions

Degut al meu perfil com a enginyer informàtic, des d'un primer moment vaig tenir clar que volia realitzar alguna aplicació amb una component gràfica o de visualització important. Tot i així no em volia estancar en una aplicació independent i prou, i tenia en ment alguna cosa més complexa que requerís algun *backoffice* de gestió accessible via web.

Vaig començar a cercar projecte de final de carrera entre les propostes que s'ofereixen a través del racó de la FIB, i quina va ser la meva sorpresa que ben aviat vaig trobar-ne un que encaixava perfectament amb les meves aspiracions. Vaig contactar ràpidament amb Borja Vallés i Alex Ballarín, que em van explicar amb més detall la proposta.

Es tractava d'una plataforma integral de gestió de museus que, a més, pretenia recrear-los en 3D per a un visitant virtual. Tenia les dues parts que a mi més m'interessaven:

- Una web de gestió amb diversos rols d'usuari per a la gestió de la plataforma, que permetia aprofitar les tecnologies d'avui en dia per centralitzar i millorar la gestió dels inventaris d'un museu.
- Una aplicació gràfica què es nodriria de les dades introduïdes en la primera part i permetria als visitants del museu fer la visita virtual en 3D de la forma més realista possible, fins i tot augmentant-ne l'experiència amb contingut multimèdia addicional.

Resultava una proposta molt atractiva i desafiant alhora, en part per la meva experiència prèvia en projectes similars i pel meu perfil d'enginyer informàtic, però es veia un volum de feina considerable en el poc temps que volia realitzar-lo.

Així doncs neix la plataforma iMuseums3D, un sistema de gestió d'inventaris i espais de museus, que permet fer visites virtuals als seus visitants.

## 2. Introducció

### 2.1 El museu tradicional

Quan es tracta de museus, tots tenim al cap una idea més o menys semblant de què estem parlant. Aquells edificis que visitem compulsivament quan anem de turisme a una ciutat, espais emblemàtics que recullen una part de la nostra cultura o història. N'hi ha que fins i tot no tracten de persones: fauna, flora, dinosaures, l'espai exterior... Però en general tots ells es caracteritzen per un espai físic on s'exposa alguna cosa.

Aquesta manera de visitar museus tan tradicional, segurament no ha canviat massa des dels primers museus. Com a molt hi hem intentat afegir una mica de tecnologia ja obsoleta a sobre com per exemple les audioguies, o afegint monitors amb contingut audiovisual al costat del material que realment s'exposa, però no deixa de ser augmentar lleugerament la mateixa experiència.

I potser el problema més greu d'aquesta manera de visitar museus, és la unidireccionalitat en què flueix la informació. Si bé és cert que és molt important que el visitant obtingui informació del museu, ja que és el motiu pel qual l'està visitant, també seria molt interessant que els gestors del museu poguessin recopilar informació dels seus visitants per tal de millorar l'experiència de la visita. Imagino que estarem d'acord que molt poca gent perd el seu temps en omplir enquestes de satisfacció anònimes després d'haver visitat un museu, i encara menys tira alguna nota a la bústia de suggeriments. Així doncs caldria trobar alguna manera anònima i implícita de recollir dades dels visitants d'un museu sense que fos invasiva o ataqués a la seva privacitat... Força complicat en el món real, sense omplir el museu de càmeres de videovigilància o dispositius de localització als visitants.

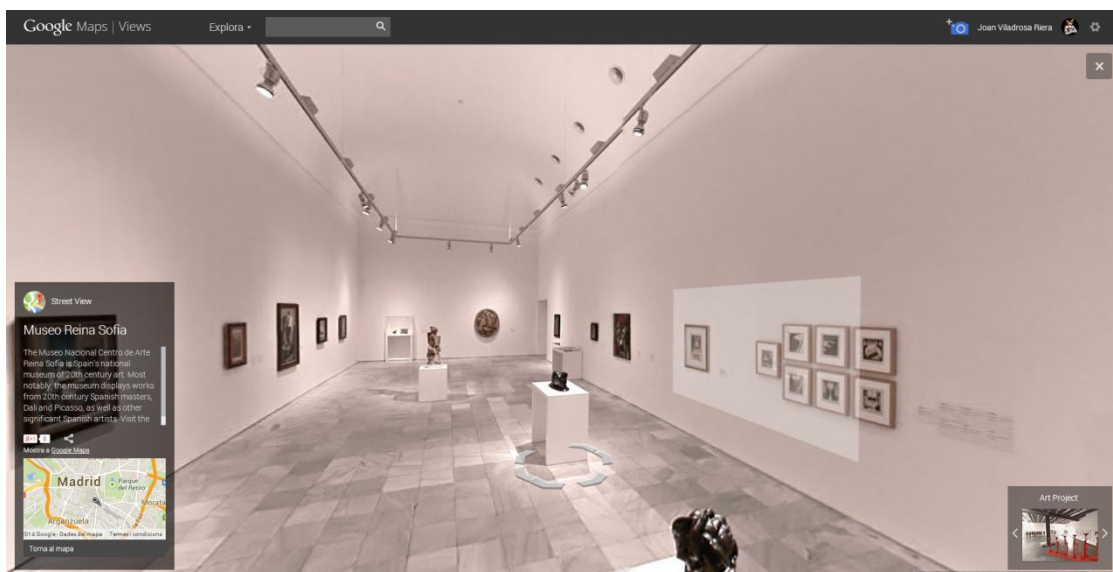
I aquí és on entra en joc la nostra proposta. Volem crear una experiència alternativa a la típica visita de museu. Volem recrear-la en un espai virtual, però afegint-hi totes aquelles coses que ens agradaria que existissin en un museu real, però que no són factibles per motius tecnològics, econòmics o físics. Està clar que de la mateixa manera que els *ebooks* mai substituiran els llibres en paper, una visita virtual en un museu mai no substituirà els museus de veritat, però factors com la comoditat, facilitat d'accés o usabilitat ens poden fer considerar aquesta alternativa com a més que interessant d'explorar.

### 2.2 Museus Virtuals a l'actualitat

Cada vegada més museus importants tenen algun tipus de visita virtual, però val a dir que no tots assoleixen amb el mateix grau d'encert una bona experiència per a l'usuari. Alguns fins i tot s'atreveixen a dir que una galeria d'imatges, com si fos un àlbum del *Facebook*, és una visita virtual. Tot i així s'ha de reconèixer que això són casos aïllats, i la majoria fan una prou bona feina.

En un cas molt semblant al mencionat anteriorment, tenim una pàgina web més estructurada, amb fotografies del museu o les obres exposades, barrejades amb text explicatiu i d'altre contingut audiovisual. Segueix sense donar-nos la sensació de visita, ja que més aviat sembla un catàleg o fulletó explicatiu de l'exposició, però en format web en lloc d'imprès en paper. Per exemple el Museu d'Història de Canadà ens proposa una visita virtual com una part més de la seva web.

Ja començant en la bona direcció, molts utilitzen sistemes de fotografies panoràmiques de 360 graus de cada sala. Això dóna una experiència bastant semblant (almenys visualment) a visitar el museu en persona, però no dóna massa llibertat de moviment a l'usuari, ja que només es pot desplaçar del centre d'una habitació a una d'altra. A més a més, és poc flexible pel que fa a la seva reusabilitat ja que per cada museu caldrà refer tota la tecnologia subjacent i anar-hi físicament amb material tecnològic i personal d'un alt grau d'especialitat. A més a més si realitzem canvis habitualment en el museu ens caldrà refer les fotografies, i això segurament té un cost força elevat, per no mencionar el maldecap d'estar contractant contínuament especialistes externs per a realitzar aquesta tasca. N'és un exemple d'aquesta aplicació el museu Thyssen.



Alguns donen un pas més enllà i utilitzen la tecnologia de *Google Street View* per a millorar la interacció i navegabilitat de l'usuari. Realment és la mateixa tecnologia anterior, però amb molts més nodes de fotografies panoràmiques que donen una sensació més continua de moviment per l'espai. Malauradament a vegades els camins no queden del tot clars i intentant navegar cap a una direcció acabem en una altra sala o planta. A més a més la deformació de perspectiva d'algunes fotos produeix zones amb força baixa resolució. Tot i així s'ha de reconèixer que és una gran millora, i els efectes visuals de moviment entre nodes fan molt creïble la visita virtual. Podem veure una gran col·lecció de museus i edificis emblemàtics que utilitzen aquesta tecnologia a la web de *Art Project* de *Google*.

Més semblant a la nostra proposta, trobem museus recreats en geometria 3D i que permeten una lliure navegació per part de l'usuari. En son un exemple molt interessant el *Museo Virtual 3D de la Informática de la Universidad de Valladolid*. Per desgràcia sembla que devia ser un projecte experimental i aïllat de fa molts anys, i que no va tenir continuïtat: utilitza tecnologia obsoleta i arcaica i la visualització final sembla més un videojoc dolent dels anys noranta que una visita a un museu de veritat. A més a més, res indica que es tracti d'algun software reutilitzable, i adaptar-ho a un altre museu o espai requeriria un gran desenvolupament.



### 2.3 Museus Virtuals en un futur

Creiem que, gràcies a les tècniques avançades de gràfics 3D en els ordinadors i altres dispositius multimèdia com smartphones o videoconsoles, el futur de la virtualització d'espais passa per la recreació en geometria 3D. Aquesta tecnologia permetria una experiència molt similar a la que s'obtidria en visitar un museu en persona, ja que permet una lliure navegació per l'espai, i infinitat de punts de vista diferents. A més a més si li sumem algun aparell tecnològic de realitat virtual, com el tan de moda avui en dia *Occulus Rift*, l'experiència tindria una immersió total.

Si bé és cert que es pot modelar en 3D cada museu al detall per a una major fidelitat, això dispararia els costos d'aplicació d'aquesta tecnologia, ja que no tots els museus es poden permetre contractar algú què els realitzi aquesta tasca. Per tant creiem que per tal que aquesta tecnologia sigui factible, cal crear un editor prou senzill d'utilitzar pels propis gestors del museu. Si aconseguim crear aquesta eina, no només tindrem museus en 3D per a fer-hi visites virtuals, sinó que el seu manteniment i actualització serà assequible i viable.

En l'era de les aplicacions web i les dades "al núvol", sembla gairebé obligat que una idea tan fortament lligada a Internet per com accedeixen els usuaris a la visita virtual, s'implementi també en un entorn web i accessible des de qualsevol punt del món. Per això, tot i que potser a algunes tecnologies de desenvolupament web els faria falta encara una mica més de refinament per a ser 100% usables, confiem que implementar aquest editor i plataforma com una "aplicació web" és anar en la direcció correcta.



## 2.4 El mercat de la plataforma

El tipus de museus que segurament veuran més beneficis en utilitzar aquesta plataforma, seran els museus més petits, que no poden fer una gran inversió en una eina personalitzada.

Així doncs, assumint certes limitacions per a la generalització, poden utilitzar un editor molt senzill que pot ser gestionat per personal no especialitzat, i com a benefici hi poden fer canvis molt més sovint de forma autònoma sense necessitat de contactar amb els administradors del sistema.

Els museus més grans i amb més pressupost, els seguirà interessant fer un producte més diferenciat i exclusiu, ja sigui per mantenir la marca i disseny propis. A més a més, segur que no tenen massa problemes per fotografiar tot el museu cada vegada que canvien l'exposició.

En la demostració del producte, i durant el desenvolupament, hem tingut en ment i usat com a exemple el museu de Can Tinturé. És un petit museu d'Esplugues de Llobregat amb la seva seu en una casa de finals del segle XIX construïda per l'arquitecte Claudi Duran i Ventosa. És el primer museu monogràfic de la rajola de l'estat espanyol. Per les característiques del museu li és ideal una plataforma de les nostres característiques: En primer lloc la geometria dels seus espais és perfectament representable amb el nostre editor. Totes les seves obres són "quadres plans", del tipus que podem mostrar. I a més el valor de l'exposició rau més en la història i contingut audiovisual addicional que en la contemplació explícita de les obres, que es podria incloure de forma òptima en el nostre visualitzador.





### 3. Proposta

En el següent apartat es desenvolupa la proposta inicial pel desenvolupament del projecte. Es defineixen els requisits bàsics, tant funcionals com no funcionals, l'arquitectura i els components tecnològics que la componen.

#### 3.1 Requisits funcionals

A l'iniciar el desenvolupament de la plataforma, vam definir una sèrie de funcionalitats que aquesta havia de tenir. Aquests requeriments van ser dividits en mòduls funcionals que corresponen a les diferents seccions que té la plataforma:

**Usuaris:**

- Invitació/creació usuaris
- Accés/identificació a la plataforma
- Usuaris multi-rol

**Museus:**

- Creació museus
- Edició de les seves dades
- Enllaços a xarxes socials
- Contingut multimèdia-audiovisual

**Personal:**

- Assignació de rols de museu a usuaris
- Gestió del personal

**Inventari:**

- Creació/modificació obres
- Afegir contingut addicional com text enriquit, imatges i vídeos

**Dissenyador:**

- Creació plantes
- Creació i posicionament de sales a les plantes
- Posicionament obres a les parets de les sales

**Navegació:**

- Navegació lliure per les plantes, sales i espais del museu
- Visualització de contingut addicional de les obres
- Enregistrament de dades estadístiques del visitant virtual

**Estadístiques:**

- Tractament i visualització de les dades estadístiques

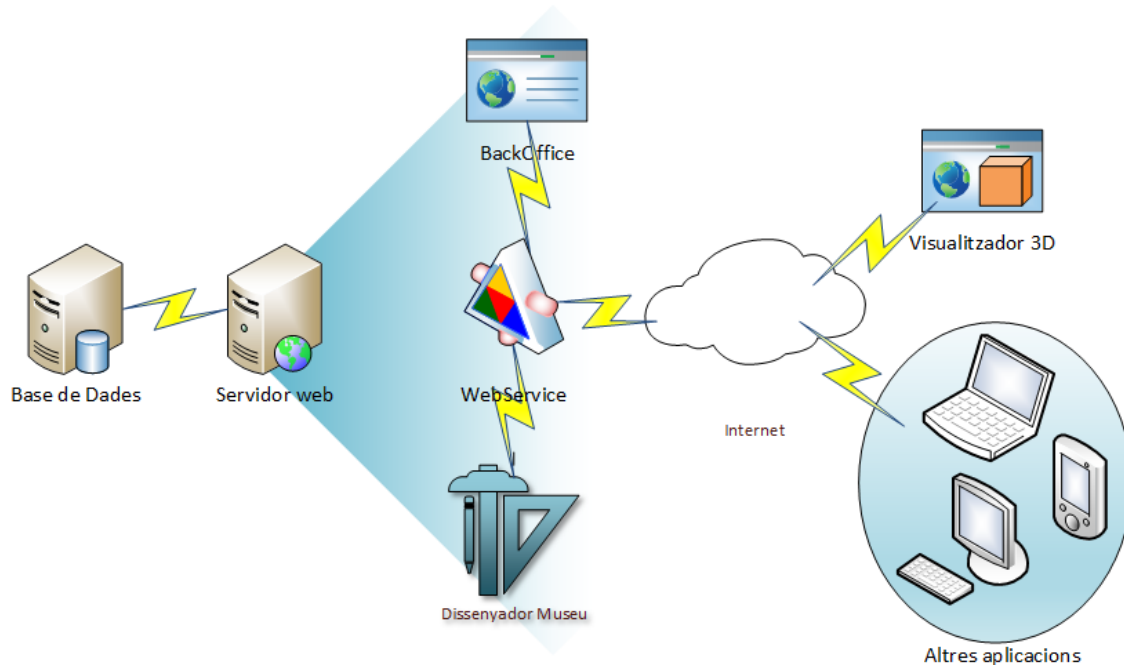
### 3.2 Requisits no funcionals

Es van definir una sèrie de requisits no funcionals bàsics per al desenvolupament del projecte. Per a la seva elaboració es van tenir en compte conceptes com l'accessibilitat, connectivitat i interoperabilitat.

- El *backoffice* serà íntegrament web i online.
- El web utilitzarà tecnologies de codi obert i estàndard.
- El web funcionarà correctament en les últimes versions disponibles dels navegadors web més utilitzats (*Internet Explorer, Google Chrome, Mozilla Firefox, Safari, Opera*).
- El visualitzador del museu 3D haurà de funcionar en navegador web.
- El visualitzador del museu 3D requerirà connexió a Internet.
- S'implementaran *APIs* públiques, tant pel funcionament intern de la plataforma, com per a la interconnexió d'aquesta amb aplicacions de tercers.
- Tota la plataforma estarà dissenyada per a una fàcil internacionalització dels textos.

### 3.3 Arquitectura del sistema

Es pretén desenvolupar una plataforma, el que fa que no tinguem una sola aplicació aïllada, sinó un conjunt de peces que encaixen entre sí per tal de crear una experiència d'usuari integrada.



**Base de Dades:** Com és normal en tota aplicació, ens caldrà una capa de persistència. Aquí és on es desaran les dades de la plataforma. Constarà d'una base de dades pròpiament dita, però també hi tindrem un sistema de fitxers on desarem, per exemple, les imatges de les obres d'art.

**WebService:** Serà la peça encarregada de comunicar la base de dades amb la resta de les parts. Fins i tot gran part del *backoffice* l'utilitzarà per tal de comunicar-se, obtenir i modificar dades de la plataforma; així evitem duplicar la mateixa funcionalitat per dues bandes.

**Backoffice:** Aquí és on es gestionarà la plataforma realment. Des dels usuaris d'administració, fins a les obres de cada museu, passant pels enllaços a xarxes socials. Més endavant veurem desglossades totes les seves funcionalitats.

**Dissenyador Museu:** Tot i que realment estarà integrat en el *backoffice*, és una part prou complexa de la plataforma com per considerar-la una peça independent. Des d'ella podrem dissenyar les sales i espais del museu i posicionar-hi les obres disponibles.

**Visualitzador 3D:** Serà l'aplicació que veurà el visitant del museu. Es nodrirà de les dades introduïdes a través del *backoffice* i crearà la representació virtual en 3D.

**Altres Aplicacions:** Gràcies al *WebService* públic, es poden crear d'altres aplicacions que obtinguin dades de la plataforma de forma senzilla. Un exemple és el projecte de final de carrera de Rubén Rodríguez Alcalde, iMuseums, una aplicació per a *smartphones Android* que pretén ser una videoguia 2.0.

## 4. Planificació

En el següent capítol es desenvolupa la planificació del projecte de forma més detallada. Aprofundirem sobre les funcionalitats del projecte definint les històries d'usuari i es detallaran de forma més específica les eines utilitzades tant en el desenvolupament com la gestió del projecte.

### 4.1 Històries d'usuari

A partir dels mòduls funcionals es generen les històries d'usuari i d'aquesta manera poder expressar els requisits funcionals de forma més col·loquial, utilitzant el llenguatge quotidià de les persones:

- **Usuaris:**
  - Com usuari vull poder invitar/crear d'altres usuaris al sistema.
  - Com usuari vull poder fer *login* al sistema.
  - Com usuari vull mantenir la sessió oberta encara que tanqui el navegador.
  - Com usuari vull poder recuperar la contrasenya.
  - Com usuari vull poder fer *logout* del sistema.
  - Com usuari vull poder utilitzar diversos rols amb unes mateixes credencials.
- **Museus:**
  - Com administrador vull poder crear museus.
  - Com a gestor vull poder modificar les dades del museu.
  - Com a gestor vull poder introduir enllaços a xarxes socials.
  - Com a gestor vull poder afegir contingut multimèdia o audiovisual al museu.
  - Com administrador vull poder eliminar museus.
- **Personal:**
  - Com a administrador vull poder assignar rols de museu als usuaris.
  - Com a administrador vull poder llistar el personal d'un museu.
  - Com a administrador vull poder llistar tots els usuaris del sistema.
- **Inventari:**
  - Com a gestor vull poder crear o modificar obres del museu.
  - Com a gestor vull poder especificar les mides de les obres.
  - Com a gestor vull poder afegir contingut addicional com text enriquit, imatges i vídeos.
  - Com a gestor vull poder cercar fàcilment entre totes les obres del museu.
- **Dissenyador:**
  - Com a dissenyador vull poder crear les diferents plantes o espais del museu.
  - Com a dissenyador vull poder crear i posicionar les sales del museu.
  - Com a dissenyador vull poder posicionar les obres a les parets de les sales.
  - Com a dissenyador vull poder connectar sales mitjançant portes.
  - Com a dissenyador vull poder posar nom i descripció a les sales.

- **Navegació:**
  - Com a visitant vull poder seleccionar el museu a visitar entre els disponibles.
  - Com a visitant vull poder navegar lliurement pels espais del museu.
  - Com a visitant vull poder obtenir informació addicional de les obres.
  - Com a visitant generaré dades estadístiques del meu recorregut i interès per les obres del museu.
- **Estadístiques:**
  - Com a gestor vull poder recuperar les dades estadístiques dels visitants.
  - Com a gestor vull poder veure les dades estadístiques agregades en indicadors o gràfics útils i representatius.

## 4.2 Disseny tècnic

### 4.2.1. Llenguatge de programació

Hem escollit utilitzar *Python* com a llenguatge de programació de la plataforma. Hi ha hagut diversos motius per escollir aquest llenguatge. En primer lloc la meua experiència prèvia en aquest entorn, tot i que amb versions més antigues del llenguatge. I després per la no experiència del meu tutor en aquest llenguatge i la seva curiositat per a veure'l en acció.



*Python* és un llenguatge molt "expressiu", és a dir, els programes són molt compactes, solen ser bastant més curts que el seu equivalent en llenguatges com C, per a molts és considerat un llenguatge de programació de molt alt nivell. És molt llegible, la sintaxis és molt elegant i permet l'escriptura de programes que resulten fàcils de llegir, en comparació amb altres llenguatges com *PHP*. Es pot utilitzar com a llenguatge imperatiu procedimental o com a llenguatge orientat a objectes, i en el nostre projecte utilitzarem els dos sistemes alternativament: imperatiu procedimental sobretot per a la capa de vistes, i orientat a objecte en el model.

### 4.2.2. Framework web

Usarem *Django*, ja que és el més popular i utilitzat en *Python*. Segueix el paradigma *Model-View-Controller* (MVC) i ens simplificarà moltes tasques del desenvolupament. Principalment ens beneficiarem de:

- Un *data mapper* que ens crearà i gestionarà la base de dades directament amb l'especificació de les classes amb *Python* corresponents.
- Un sistema de gestió de *URLs* basat en expressions regulars que ens cridarà les funcions corresponents del domini de l'aplicació.
- Un sistema de vistes basat en *templates*. Els *templates* són molt rics en funcionalitat, suportant per exemple herència de blocs, inclusió d'altres *templates*, blocs condicionals o bucles o fins i tot afegir *tags* que executen codi propi.
- Sistema de *login* d'usuaris. *Django* ens proporciona un sistema bàsic d'autenticació i control d'usuaris. Tot i que l'haurem d'ampliar, ens servirà com a base i sobretot oblidar-nos d'implementar el sistema de *cookies* i permisos.

- Sistemes de seguretat. Disposem de moltes mesures de seguretat contra atacs típics en llocs web ja implementades. Les més destacades: *Cross site Scripting* (XSS), *Cross site request forgery* (CSRF), *SQL Injection*, *User-Uploaded content*. Ara no entrarem en detall a explicar cadascuna d'elles. Per a més informació consulteu el glossari.

# django

## 4.2.3. Entorn de treball web

L'entorn de desenvolupament integrat (o IDE en les seves sigles en anglès) *PyCharm* de *JetBrains*, és un dels més recomanats i populars per al desenvolupament de *Python* conjuntament amb *Django*.

Té una autocompleció de codi molt espectacular tenint en compte que es tracta d'un llenguatge interpretat i poc tipat. Suporta la interpretació de la sintaxis dels principals llenguatges de programació web: *Javascript*, *HTML*, *CSS*. També disposarem d'un *debugger* fantàstic, integració amb *git* i molt més!

Tot i que disposa d'una versió *OpenSource* gratuïta, no té totes les funcionalitats que necessitem actives, així que malauradament haurem d'adquirir alguna de les versions de pagament. Per sort hi ha disponible una versió per a ús acadèmic que ens encaixa perfectament amb el nostre ús, i que a més té un preu molt reduït.



Commercial License	Personal License	Academic License	Classroom License	Open Source Project License
For companies and organizations <a href="#">Learn more »</a>	For individual developers <a href="#">Learn more »</a>	For students and teachers <a href="#">Learn more »</a>	For trainers and educational institutions <a href="#">Learn more »</a>	For open source projects <a href="#">Learn more »</a>
€179 + VAT <a href="#">Buy now</a>   <a href="#">Get quote</a> <a href="#">50% OFF for startups »</a>	€89 + VAT <a href="#">Buy now</a>	€26 + VAT <a href="#">Apply now</a>	FREE <a href="#">Apply now</a>	FREE <a href="#">Apply now</a>
Upgrade Subscription Renewal <a href="#">Learn more »</a>	Upgrade Subscription Renewal <a href="#">Learn more »</a>	Upgrade Subscription Renewal <a href="#">Learn more »</a>		
€89 + VAT <a href="#">Renew now</a>	€53 + VAT <a href="#">Renew now</a>	€17 + VAT <a href="#">Renew now</a>		

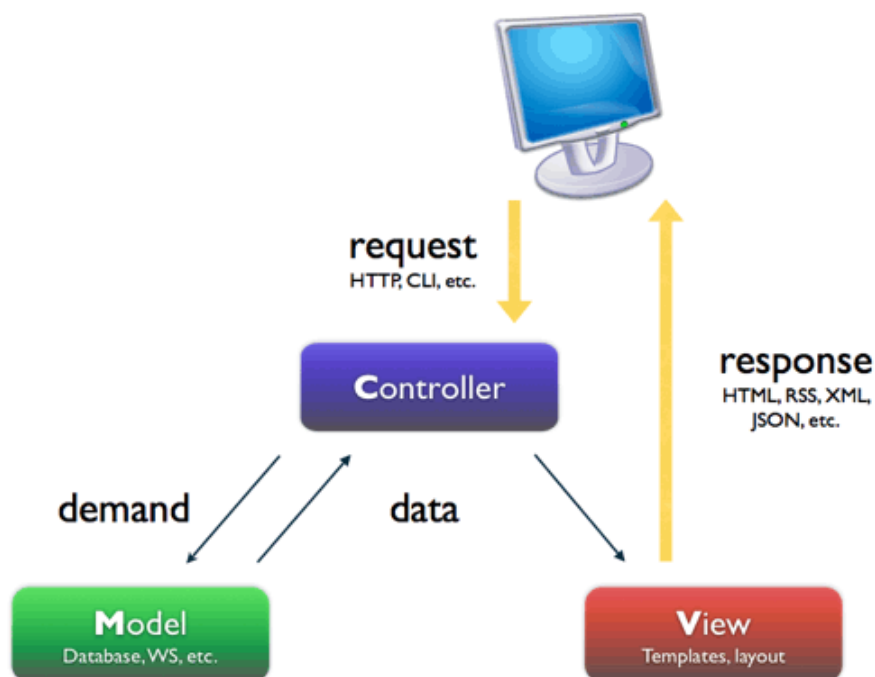


#### 4.2.4. Patró Model-Vista-Controlador

El model–vista–controlador (MVC) és un patró d'arquitectura de software que separa les dades i la lògica de l'aplicació de la interfície d'usuari i el mòdul encarregat de gestionar els events i les comunicacions. Per a tal finalitat MVC proposa la construcció de tres components diferenciats que són el model, la vista i el controlador, és a dir, per una banda defineix components per a la representació de la informació, i per l'altra la interacció amb l'usuari. Aquest patró d'arquitectura de software es basa en les idees de reutilització de codi i separació de conceptes, característiques que busquen facilitar la tasca de desenvolupament d'aplicacions i el seu posterior manteniment.

Concretament en una aplicació web tindrem:

- **Model:** És principalment la base de dades, i qui guardarà la persistència de l'aplicació. En el nostre cas també inclouria el conjunt de classes que serveixen per accedir més còmodament a les dades (*data mapper*).
- **Controlador:** És la lògica de l'aplicació. S'encarregarà de rebre les peticions HTTP de l'usuari, identificar-les i cridar el mètode corresponent. Aquest consultarà la capa del model, operarà amb les dades i farà les modificacions corresponents. Finalment cridarà a la capa de vista
- **Vista:** Són els *templates* i d'altres documents que es visualitzaran en el navegador web de l'usuari. Generalment seran *HTML*, però també podran ser *JSON* (per a les peticions *AJAX* per exemple) o d'altres llenguatges. Mostraran a l'usuari la interfície amb que interactuarà.



#### 4.2.5. Patró Data Mapper

El patró *data mapper* ens permet interactuar amb una base de dades, sense necessitat de ser conscients de la seva implementació o detalls de representació. Simplifica els accessos mitjançant interfícies amb el mateix llenguatge de programació que la resta de l'aplicació, en lloc de consultes complexes amb el llenguatge de la base de dades.

Aquestes interfícies tindran els mètodes de crear, llegir, modificar i eliminar, i només amb aquestes operacions ja tindrem tota la funcionalitat que necessitem.

#### 4.2.6. Mapatge d'objectes relacional

Molt relacionat amb el punt anterior tenim el mapatge d'objectes relacional (ORM, sigles en anglès de *Object-relational mapping*). És una tècnica de programació per convertir dades de llenguatges de programació orientats a objectes en la seva representació en bases de dades relacionals, a través de la definició de les correspondències entre els diferents sistemes.

En el nostre cas només definint l'estructura de dades amb *Python* i unes anotacions sobre els tipus de dades, el *framework* de *Django* ja s'encarregarà de fer la conversió a *MySQL*.

#### 4.2.7. Internacionalització

*Django* ens proporciona un sistema molt còmode d'internacionalització dels textos. En els *templates* i altres llocs de l'aplicació on utilitzem literals de text que s'haurien de traduir, utilitzarem el *tag* `{% trans "text" %}`. Podem posar el text per defecte a mostrar directament en la llengua per defecte de la nostra pàgina web. Després executant el *script* de *manage.py* de *Django* ens auto generarà els fitxers de text a traduir. Llavors només ens caldrà modificar aquests fitxers amb els textos en cada idioma, i el *framework* de forma automàtica agafarà els textos de cada idioma.



#### 4.2.8. HTML5 + CSS3 + Bootstrap + jQuery

La interfície gràfica d'usuari de la nostra aplicació, serà íntegrament web. Per la part del servidor ja hem vist que utilitzarem els *templates* de *Django*, però en la part del client, utilitzarem un bon munt de tecnologies molt habituals:



- **HTML5** (acrònim d'*Hyper Text Markup Language*), és un llenguatge de marcat que deriva de l'*SGML* dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents per a la web. La nova versió 5 estableix una sèrie de nous elements i atributs que reflecteixen l'ús típic dels llocs web moderns. Alguns d'ells són tècnicament similars a les etiquetes `<div>` i `<span>`, però tenen un significat semàntic, com ara `<nav>` (bloc de navegació del lloc web) i `<footer>`. Altres elements proporcionen noves funcionalitats a través d'una interfície estandarditzada, com els elements `<audio>` i `<video>`. També presenta millores en l'element `<canvas>`.
- **CSS3** és un llenguatge de fulls d'estil utilitzat per descriure la semàntica de presentació (l'aspecte i format) d'un document escrit en un llenguatge de marques. La seva aplicació més comuna és dissenyar pàgines web escrites en *HTML*. Està dissenyat principalment per permetre la separació de contingut del document (escrit en *HTML* o un llenguatge de marques similar) de la presentació del document, incloent-hi elements com la disposició, colors, i fonts. Aquesta separació pot millorar l'accessibilitat al contingut, proporcionar més flexibilitat i control en l'especificació de característiques de presentació, permetre que múltiples pàgines comparteixin un format comú, i reduïx complexitat i repetició en el contingut estructural (com per exemple al permetre disseny web sense taules). El nivell de CSS 3 és actualment en desenvolupament, des del 15 de desembre de 2005. A diferència de CSS2, que va ser una especificació que definia diverses funcionalitats, CSS3 està dividida en diversos documents separats, anomenats mòduls. Cada mòdul afegeix noves funcionalitats a les definides a CSS2, de manera que es preserven les anteriors per mantenir la compatibilitat. Degut a la modulació dels CSS3, diversos mòduls poden trobar-se en diferents estadis de desenvolupament.
- **Bootstrap** és un *framework* o conjunt d'eines de software lliure pel disseny de llocs i aplicacions web. Conté plantilles de disseny amb tipografia, formularis, botons, requadres, menús de navegació i altres elements de disseny basats en

*HTML* i *CSS*, així com extensions de *Javascript* opcionals addicionals. És el projecte més popular a *GitHub*, i és usat en infinitat de llocs, destacant per exemple la *NASA*, la *MSNBC* o la pròpia *Twitter*. *Bootstrap* és modular i consisteix essencialment en una sèrie de fulles d'estil *LESS* que implementen la varietat de components de l'eina. Aquest llenguatge permet l'ús de variables, funcions i operadors, selectors aniuats, així com classes *mixin*. A més a més és compatible amb les últimes versions dels principals navegadors, i es degraden alguns afegits estètics amb navegadors més antics com Internet Explorer 8 (per exemple les cantonades arrodonides).

- **jQuery** és una biblioteca de *Javascript*, creada inicialment per *John Resig*, que permet simplificar la manera d'interactuar amb els documents *HTML*, manipular l'arbre *DOM*, manejar events, desenvolupar animacions i afegir interacció amb la tècnica *AJAX* a pàgines web. Va ser presentada el 14 de gener de 2006 en el *BarCamp NYC*, i és la biblioteca *Javascript* més utilitzada. Ofereix una sèrie de funcionalitats que d'una altra manera requeririen molt més codi, és a dir, amb les funcions pròpies d'aquesta biblioteca s'obtenen grans resultats en menys temps i espai. La majoria de grans empreses com *Google*, *Microsoft* o *Nokia*, l'utilitzen en les seves plataformes o sistemes web.

#### 4.2.9. Motor Visualitzador 3D

Per al desenvolupament del nostre visualitzador 3D hem escollit *Unity3D*. *Unity* és un motor de videojocs multiplataforma creat per *Unity Technologies*. Està disponible com a plataforma de desenvolupament per a *Windows* i *Mac OS X*, i permet crear jocs per a *Windows*, *OS X*, *Linux*, *Xbox 360*, *Playstation 3*, *Playstation Vita*, *Wii*, *Wii U*, *iPad*, *iPhone*, *Android* i *Windows Phone*. Gràcies al *plugin* web de *Unity* també es poden desenvolupar jocs de navegador per a *Windows* i *Mac* (tot i que no està suportat oficialment també es poden executar en *Linux*).



El principal avantatge d'utilitzar aquesta tecnologia és el desplegament en múltiples plataformes. Tot i que originalment només ens interessa la versió per a navegador web, amb molt poc treball addicional (de fet només cal seleccionar la plataforma d'una llista i clicar a "*build*") podríem crear versions per a la infinitat de dispositius i sistemes que hem mencionat abans. A més a més, per a desenvolupament en navegador web és la millor opció, ja que té un rendiment molt més elevat que d'altres tipus de *plugins* com *Adobe Flash*, i ja no parlem de codi interpretat amb *Javascript+HTML5*.

El motor de *Unity3D*, inclou a més un entorn de desenvolupament complert, amb un editor molt ric i funcional, tant d'escenes 3D com de codi. Tot i així nosaltres realment no utilitzarem de forma extensiva aquestes funcionalitats, ja que tota la geometria 3D la crearem via codi en funció del que rebem de la plataforma, i com a *IDE* utilitzarem *Visual Studio*, què és encara més còmode i funcional.

#### 4.2.10. Visual Studio C# Express

El llenguatge de programació que utilitzarem amb *Unity* és *C#* (tot i que també podríem utilitzar *Javascript* o *Boo*), i no hi ha millor *IDE* per a aquest llenguatge que *Microsoft Visual Studio*. A més a més, *Microsoft* ens posa a la seva disposició de versions "express" gratuïtes, que tot i que estan força més limitades en funcionalitat ens són suficients per a la seva integració amb *Unity3D*.



## 4.3 Metodologia de Treball

### 4.3.1. Control de versions de codi

En tot projecte de mitjana o llarga duració, és necessari utilitzar un sistema de control de versions. En aquest s'enregistren els canvis que es realitzen sobre els elements del producte, i així podem tenir un millor control de l'estat del projecte en qualsevol moment. Són especialment útils sobretot en equips, per poder compartir i modificar el codi simultàniament sense interferir en el treball de la resta de programadors. Tot i que en aquest projecte només hi haurà un sol programador, i no utilitzarem aquesta característica.

Ens hem decidit per utilitzar git, en concret la popular forja *GitHub*. Aquesta ens proporciona, a més, una interfície web amigable per a gestionar el servei, i a més informació estadística útil en forma de gràfics. Normalment, els repositoris són públics, el que significa que qualsevol persona d'Internet pot accedir al codi font, i fins i tot bifurcar-lo en un altre repositori per tal de continuar un desenvolupament paral·lel. Aquesta característica no és pas gens desitjable durant el desenvolupament del nostre projecte de final de carrera, però per sort, i gràcies a que som estudiants d'universitat, ens proporcionen repositoris privats de forma totalment gratuïta.



### 4.3.2. Servidor de producció

En projectes web és habitual que el desenvolupador corri en la mateixa màquina en la qual desenvolupa el projecte un servidor on s'executarà el software. Això és molt útil per *debugear* el software, per reduir temps de càrrega de les pàgines (a l'executar-se en local no tenim temps de latència degut a la xarxa) i per estalviar costos de servidors.

Tot i així, ens cal algun lloc públic on poder penjar el projecte per tal d'ensenyar-lo a la resta d'usuaris del món. Com veurem més endavant, en aquest projecte hem utilitzat metodologies àgils de desenvolupament, i un seguiment molt constant i freqüent amb el tutor. Això, sumat a les nombroses reunions via *Skype*, ha provocat la necessitat de tenir un servidor "de producció" per poder mostrar els avenços realitzats a cada iteració.

Hem decidit utilitzar els serveis web de *Amazon*, sobretot per la seva fàcil escalabilitat i flexibilitat d'ús. *Amazon* proporciona per a noves comptes, un seguit de serveis gratuïts durant un any, que tot i ser una mica limitats per a projectes grans, permeten una més que suficient potència per a la fase de desenvolupament.

Concretament hem utilitzat tres serveis:

- **Amazon Elastic Compute Cloud (Amazon EC2)** és un servei web que proporciona capacitat informàtica amb mida modificable al núvol. Està dissenyat per a facilitar recursos escalables basats en web. És a dir, seran el nostre servidor virtual. En ell s'executarà el servidor web, i hi hauran desats els fitxers del nostre projecte.
- **Amazon Relational Database Service (Amazon RDS)** és un servei web que facilita les tasques de configuració, operació i escalat d'una base de dades relacional en el núvol. Proporciona capacitat rentable i de mida modificable i, al mateix temps, gestiona les tedioses tasques d'administració de la base de dades. És a dir, ens proporcionarà la base de dades relacional (*MySQL*) necessària per a la persistència de dades del nostre projecte.
- **Amazon Route 53** és un servei web *DNS* (Sistema de noms de domini) escalable i d'alta disponibilitat. Permet una senzilla configuració i integració amb la resta de serveis de *Amazon Web Services*. Bàsicament s'encarregarà de resoldre les peticions del nostre domini i dirigir-les cap al nostre servidor.



#### 4.3.3. Metodologies àgils de desenvolupament

El meu tutor de projecte, Alex Ballarín, em va proposar utilitzar *Scrum*, una metodologia de desenvolupament àgil. Està especialment indicat per a projectes on es necessiti obtenir resultats aviat, encara que siguin parcials, i els requisits són canviants o poc definits.

En primer lloc, explicarem els diferents rols d'aquesta metodologia:

- El **Product Owner** representa la veu del client. S'assegura que el resultat del treball de l'equip de *Scrum* s'adequa des de la perspectiva del negoci. El *Product Owner* escriu històries d'usuari, les prioritza, i les col·loca en el *Product Backlog*.
- El **Scrum Master** s'assegura que el procés *Scrum* s'utilitza com cal i que es compleixin les seves regles. No és el líder de l'equip (ja que aquest s'auto-organitza), sinó que actua com una protecció entre l'equip i qualsevol influència que el distregui.
- **L'equip de desenvolupament** té la responsabilitat de lliurar el producte amb el resultat dels *Sprints*.



Si bé aquests rols de treball no estaven clarament definits, degut a la dimensió del projecte i equip de treball (només érem dues persones de forma activa), els dos primers rols eren duts a terme principalment pel meu tutor, mentre que l'equip de desenvolupament era jo mateix. Tot i així jo també vaig participar activament en l'anàlisi de requisits i visió del producte.

A continuació detallarem els passos de la metodologia:

- El primer que es fa és definir els objectius del projecte i separar-los per tasques a realitzar. Llavors el *ProductOwner* les ordena per ordre d'importància i preferències, tot creant el que es coneix com a **Product Backlog**, què és, en resum, el conjunt de requeriments funcionals que defineixen la feina a fer.
- Un cop es té el *Product Backlog* redactat, l'equip fa una reunió per a planejar i prioritzar tasques. És en aquest moment quan es fa la planificació de *sprints*. Els **sprints** són períodes de temps fix d'una durada definida (poden ser de fins a quatre setmanes, però en el nostre cas eren de dues, degut a la curta duració del projecte) on l'equip realitza les tasques que tenia assignades al *Product Backlog*.
- Una vegada finalitzat cada *Sprint*, es duu a terme el **Sprint Review Meeting**, on ens reuneix l'equip per revisar la feina què s'ha pogut completar i la què no, així com una revisió del temps necessari pel seu assoliment. A més a més s'acompanya d'una petita demostració del producte en la seva iteració actual.
- Per acabar tenim el **Sprint Retrospective**, on s'opina sobre com ha funcionat l'equip durant el *Sprint*. L'objectiu de la reunió és identificar els aspectes positius i els que s'han de millorar per optimitzar el rendiment de l'equip en els següents *Sprints*.



#### 4.4 Pressupost Inicial

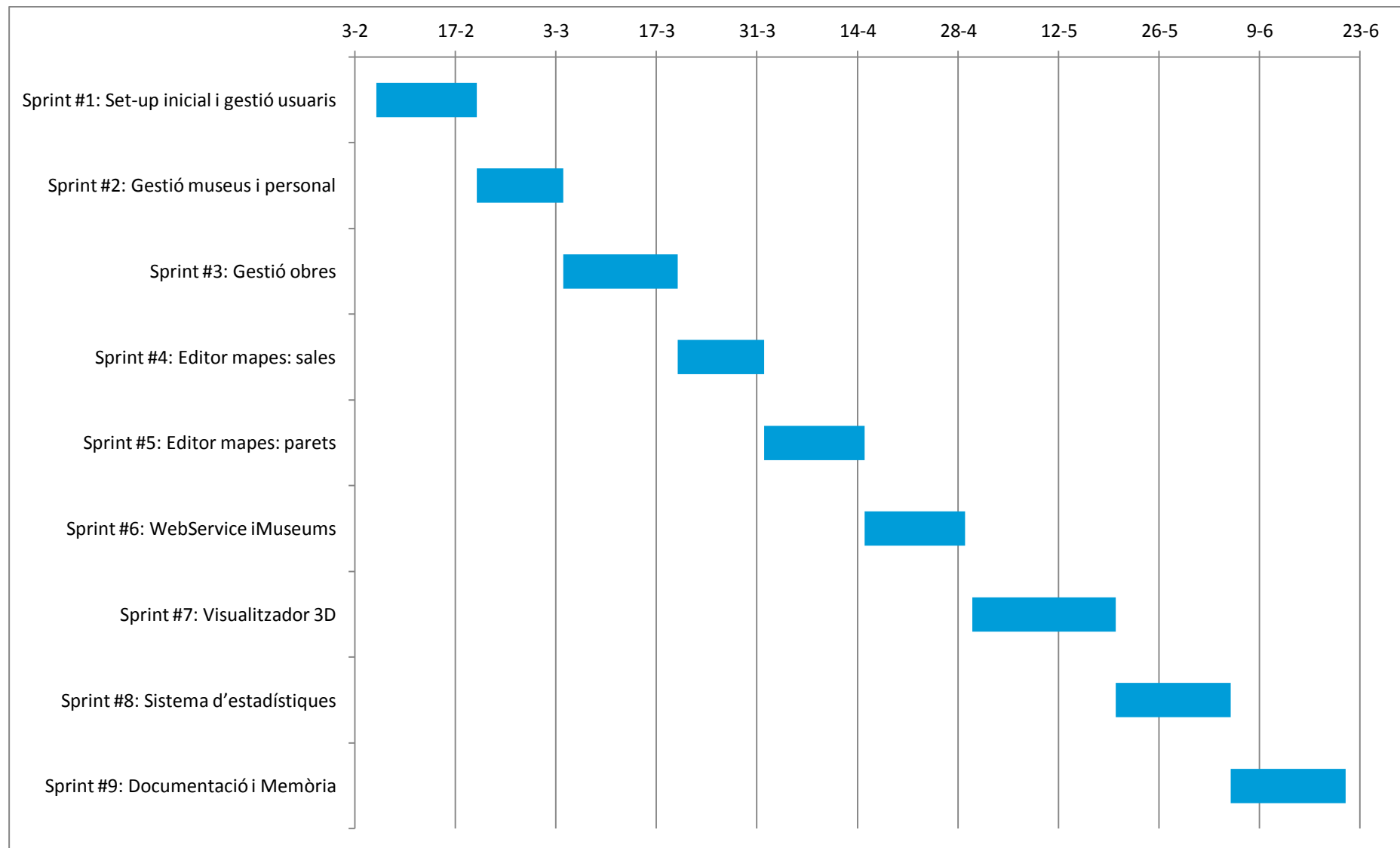
Una vegada definits els requisits bàsics i dissenyar l'arquitectura de la plataforma, passem a realitzar un pressupost del projecte. Com ja hem vist, dividirem el projecte en *sprints*, al final de cadascun dels quals hi haurà una entrega parcial de resultats. En ella hi haurà una avaluació de la feina realitzada i un balanç de l'estat del projecte actual, es planificarà el següent *sprint* i es proposaran noves millores i funcionalitats pel producte.

A continuació es detallen les hores previstes per a cada *sprint* per a realitzar les tasques assignades a cadascun d'ells:

Sprint	Tasca Principal	Hores
#0	Planificació inicial	40
#1	Set-up inicial i gestió usuaris	60
#2	Gestió museus i personal	55
#3	Gestió obres	55
#4	Editor mapes: sales	55
#5	Editor mapes: parets	55
#6	WebService iMuseums	80
#7	Visualitzador 3D	70
#8	Sistema d'estadístiques	70
#9	Documentació i Memòria	80
<b>Total hores</b>		<b>620</b>
	Cost laboral per hora	30 €
	Cost laboral total	18.600 €
	Llicència PyCharm Acadèmica	26 €
	Subtotal	18.626 €
	Marge comercial i de contingència(15%)	2.790 €
<b>Total</b>		<b>21.416 €</b>

Cal recordar que aquestes són les hores previstes de les tasques, i amb les que elaborem el pressupost inicial. Com que ja tenim en compte que podríem sofrir algunes lleugeres desviacions, per això hi afegim el marge de contingència. El preu final és el pressupost que passàrem al client o inversor, no el cost real del producte. Aquest el calcularem una vegada desenvolupat el projecte, i el podem trobar a les conclusions de la memòria.

Respecte al que fa a la planificació temporal, cada *sprint* té una durada aproximada de dues setmanes variant lleugerament depenent de la disponibilitat dels membres de l'equip. A la següent pàgina es detalla amb un diagrama de Gantt la successió de *sprints*. Com que les tasques s'han repartit seqüencialment i sense més dependències que les ja implícites en cada *sprint*, tenim un diagrama força senzill.



## 5. Desenvolupament

En el següent capítol es descriurà com s'ha dut a terme el desenvolupament i com ha anat evolucionant el projecte durant el seu transcurs. Detallarem per cada Sprint les tasques dutes a terme, la seva planificació inicial, com s'han implementat, i si és el cas, les millores i noves funcionalitats que s'han anat afegint.

Un altre punt important d'aquest capítol és mostrar l'origen i evolució tant del diagrama de classes com de l'API del *webservice* públic durant l'evolució del projecte. Només anirem mencionant les taules, classes o mètodes que s'hi afegeixen, tindrem la seva especificació detallada i completa al final d'aquest capítol i als annexos.

### 5.1 Sprint #1: Set-up inicial i gestió usuaris

- **Reunió Sprint Planning**

1. **Set-up servidors Amazon i repositori GitHub (7 hores)**

Per tal de poder començar a desenvolupar el projecte, ens caldrà posar en marxa primer la infraestructura necessària per executar i gestionar el codi. Pels servidors utilitzarem els serveis web de *Amazon*. No hi entrarem en detall, ja que els hem explicat anteriorment a l'apartat de metodologia de treball.

Aquí instal·larem i configurarem també el programari necessari en el servidor:

- **Servidor web:** Hem escollit el popular servidor web Apache. Ens caldrà configurar el virtual host de la nostra aplicació.
- **Intèrpret de scripts:** Com a llenguatge de programació de la plataforma, hem triat *Python 3.4* Tot i que la comunitat d'aquest llenguatge encara està molt estancada en l'anterior versió 2.7, i algunes llibreries encara no estan disponibles per a la nova versió o no tenen suport, són força els canvis de rendiment i productivitat què ens motiven per a arriscar-nos amb l'última versió. Inicialment instal·larem les llibreries bàsiques per a desenvolupar un projecte web, i a mesura que les anem necessitant anirem instal·lant la resta. Per començar ens cal *Django* (el framework web) i el connector amb la base de dades *MySQL*. Volíem utilitzar l'última versió de les llibreries, però el fet d'ajuntar-ho amb *Python 3*, sembla que ho converteix en odissea: el connector *MySQLdb* que utilitza *Django*, no està disponible en aquesta última versió del llenguatge. Trobem un substitut, *pyMySQL*, però és un client integrament escrit en *python* (no codi natiu) i el seu rendiment és força dolent. El connector oficial de *MySQL* en *python* només suporta fins la versió anterior de *Django*, així que tampoc ens serveix, tot i que al blog de l'autor comenta que ben aviat donarà suport per a l'última versió també. Així que de moment utilitzarem *pyMySQL* i a mesura que el projecte vagi avançant mirarem si ja podem utilitzar el connector oficial.

També crearem el repositori a *GitHub* i hi pujarem l'esquelet inicial del projecte.

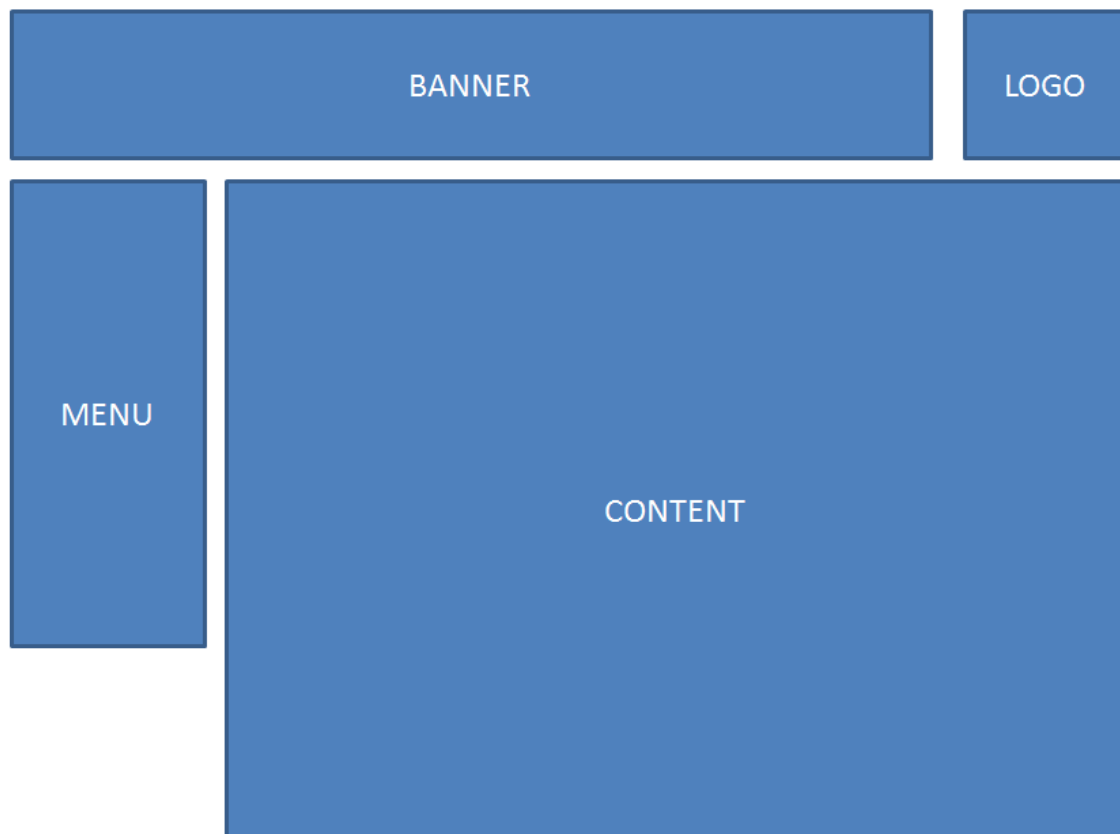
Com a *IDE* utilitzarem *PyCharm* de l'empresa *JetBrains*, com ja hem comentat al capítol de disseny tècnic, juntament amb els seus avantatges i beneficis, apart del seu reduït cost.

## 2. Dissenyar plantilla base web (15 hores)

En primer lloc dissenyarem la maqueta de la nostra pàgina web amb *Powerpoint*. No és una eina molt potent en aquest sentit, però ens servirà per a fer-nos una idea bàsica de com estructurar els elements a la pàgina.

A continuació implementarem, ja en *HTML5* i utilitzant la llibreria *Bootstrap 3*, la plantilla base. Aquí també hem decidit utilitzar les últimes versions de les tecnologies, ja que, quan es tracta de tecnologies web, és molt interessant usar els estàndards més moderns per no quedar-nos antiquats abans de finalitzar el projecte. Bootstrap ens proporciona un disseny sensible i ajustable a l'amplada de la pantalla, d'aquesta manera la nostra web es visualitzarà correctament en un ampli ventall de resolucions diferents.

Finalment dissenyarem les primeres pantalles necessàries en la plataforma: *Login* i *Logout* del sistema. També crearem aquí les bases de dades necessàries per al sistema d'autenticació d'usuaris de *Django*.



### 3. Formulari de ABM museu (12 hores)

Començarem per implementar la unitat de segmentació principal de la nostra plataforma: el museu. Una taula per llistar els museus del sistema, un formulari per a crear-ne de nous o modificar els existents i la possibilitat d'eliminar-los.

Afegim al *webservice* els mètodes llistar museus, obtenir museu, crear museu, editar museu, eliminar museu.

Pel que fa a la base de dades hi afegim la taula museum.

### 4. Formulari de ABM rols de museu (16 hores)

Una vegada tenim els museus, ens cal poder-hi afegir personal. En aquest cas, l'únic personal que ens interessarà des del punt de vista de la plataforma, són els que poden fer-hi *login*. Per tant, és aquí on assignarem els gestors de museu. En afegir un usuari nou, o afegir-li un rol nou a un usuari ja existent, el notificarem via *email*, enviant-li la contrasenya inicial si fos necessari.

Afegim al *webservice* els mètodes llistar rols museu, nou rol museu, eliminar rol museu.

A la base de dades hi afegim la taula role.

Museu	Usuari	Nom	Rol	
Can Tinturé	gestor@educabits.com	gestor	Gestor	<input type="checkbox"/>
	alex.ballarin@gmail.com	Alex Ballarin	Gestor	<input type="checkbox"/>
	joanviladrosa@gmail.com	Joan Viladrosa Riera	Gestor	<input type="checkbox"/>
Museu Test	alex.ballarin@gmail.com	Alex Ballarin	Gestor	<input type="checkbox"/>
	joanviladrosa@gmail.com	Joan Viladrosa Riera	Gestor	<input type="checkbox"/>

Afegir Gestor

### 5. Formulari de perfil d'usuari (10 hores)

Finalment en aquest *sprint* implementarem un formulari on l'usuari podrà canviar les seves dades personals, així com la seva contrasenya. Més endavant afegirem més dades al perfil de l'usuari, però de moment ens serveix amb aquestes per poder seguir endavant amb la resta.

- **Reunió Sprint Review**

- Es detecta un *bug* on un gestor pot visualitzar tots els museus del sistema, i no només el seu.

## 5.2 Sprint #2: Gestió museus i personal

- Reunió Sprint Planning

1. Usuaris multi-rol (4 hores)


Com els nostres usuaris seran multi-rol, els haurem de donar un mecanisme per poder seleccionar el rol actiu. Afegim a la part superior del menú, un selector de rol. Això implicarà alguns canvis a la base de dades per saber quin rol té actiu cada usuari.

2. Perfil d'usuari amb foto i enllaços a xarxes socials (4 hores)

Afegirem al formulari del perfil d'usuari, la possibilitat de pujar una imatge de perfil, així com enllaços a les xarxes socials més conegudes (*Facebook*, *Twitter*, *Google+*, *Linkedin* i *Instagram*).

### Dades personals

Imatge



Canviar Imatge

Nom

Admin


Cognoms


Cognoms


Nom Curt


Nom Curt


Xarxes Socials

 /facebook

 @twitter

 +google

 LinkedIn

 Instagram

Desar

### Canviar contrassenya

Actual

Nova

Canviar



### 3. El nom de l'usuari serà un enllaç al seu perfil (6 hores)

Volem que sempre que es faci referència al nom d'un usuari (per exemple en els llistats de rols d'un museu), sigui un enllaç a una pàgina amb informació del seu perfil. Dissenyarem la pàgina del perfil, i també haurem de modificar les vistes ja creades anteriorment per tal que el nom d'usuari enllaci amb ella. També afegirem al formulari d'edició de perfil la possibilitat d'escollir un nom d'usuari únic per a mostrar-lo a la URL del perfil, en lloc de la ID numèrica d'usuari.

### 4. Crear vistes de root (10 hores)

Afegirem un super-usuari per sobre dels administradors. De fet l'única tasca d'aquest serà crear administradors, ja que fins ara s'havien d'introduir manualment a la base de dades. Podem tenir un únic super-usuari o introduir-los manualment, però el rol d'administrador és força més comú.

Reutilitzem els *webservice* creats anteriorment per a rols, però hi afegim àlies, per tal d'accedir directament a un tipus de rol concret: administrador.

Usuari	Nom	Rol
admin@educabits.com	Admin	Administrador
alex.ballarin@gmail.com	Alex Ballarin	Administrador

Afegir Administrador

### 5. Edició informació museu (15 hores)

El gestor del museu podrà modificar les dades que abans ha introduït l'administrador, i a més d'altres, com els enllaços a les xarxes socials o pujar contingut multimèdia.

Utilitzarem la llibreria *jquery-fileupload* per fer la gestió asíncrona de pujada de fitxers, i *bootstrap-gallery* per a la galeria d'imatges.

A més a més aquí també dissenyarem la pàgina de perfil de museu, anàloga a la de perfil d'usuari.

Modifiquem la taula de la base de dades museum i creem la nova taula museum\_lang per tal de desar els camps multi-idioma del museu i museum\_media per desar-hi el contingut audiovisual.

### 6. Webservice informació museus (10 hores)

Modifiquem la primera versió inicial d'obtenir museu, afegint-hi totes les noves dades, els camps multi-idioma i el contingut audiovisual.

## 7. Posar capçalera i logo (1 hora)

Acabem el disseny base de la plantilla amb una capçalera i logo de la plataforma.



Joan Viladrosa Riera

## 8. Revisar JSON i estructura de fitxers (5 hores)

Aquí fem una primera aproximació al format i estructura de fitxers que utilitza un projecte de final de carrera paral·lel per desar la informació dels museus. Mirem que podem aprofitar, tenint en compte les limitacions del format existent. Finalment decidim que mantindrem compatibilitat amb aquest format, però l'hi afegirem més funcionalitats.

- **Reunió Sprint Review**

- Les imatges en el servidor de producció no es pugen correctament.
- Els enllaços a les xarxes socials no acaben de ser del tot intuïtius d'utilitzar. Utilitzarem un sistema que *parseja* el nom d'usuari a partir de la URL complerta al perfil de la xarxa social.
- Falta poder afegir vídeos al contingut audiovisual del museu.
- Respecte al format i estructura de fitxers per desar la informació dels museus, implementarem el format de l'altre projecte de final de carrera per tal que aquesta aplicació es pugui nodrir dels continguts de la nostra plataforma, però per una altra banda utilitzarem un nou format que suporti totes les noves funcionalitats que volem afegir al format (com per exemple sales poligonals).

## 5.3 Sprint #3: Gestió obres

### • Reunió Sprint Planning

#### 1. Temes pendents Sprint anterior (3 hores)

Al tractar-se de temes petits, mirem de solucionar els problemes detectats a l'anterior iteració en aquesta. Això inclou:

- Instal·lar les llibreries de manipulació d'imatge en el servidor de producció
- Modificar *parsing* dels enllaços a les xarxes socials a través de la URL

#### 2. Afegir vídeos al contingut multimèdia del museu (10 hores)

Modifiquem el formulari per a pujar imatges, per a poder-hi posar també enllaços a vídeos. No pujarem vídeos a la plataforma, sinó que utilitzarem enllaços a dos populars serveis de video-streaming: *YouTube* i *Vimeo*.

Haurem d'utilitzar les API d'aquests serveis per aconseguir *previews* dels vídeos per mostrar en la galeria d'imatges.

Modifiquem la taula `museum_media` de la base de dades per tal que suporti també vídeos.



### 3. Gestió de les obres (30 hores)

I acabem el sistema de gestió del museu amb la part d'inventari d'obres. Tindrem el formulari de creació/edició i el llistat.

Pel formulari reutilitzarem algunes parts de codi anteriors:

- El sistema de pujar imatge de perfil per a pujar la imatge de l'obra. Tot i que aquí desarem la imatge original apart de la miniatura, per a mostrar-la al màxim de la seva resolució en el visualitzador 3D
- El sistema d'afegir contingut audiovisual i galeria del museu.

Com a novetat, afegim la llibreria *TinyMCE* per a editar text enriquit en la descripció de les obres (que a més serà multi-idioma).

Pel llistat de les obres, ens trobem per primera vegada en un cas on segurament tindrem una llista molt extensa, i necessitarem paginar-la, ordenar-la per diferents criteris o poder-la filtrar. Utilitzarem la llibreria *Datatables*, un *plugin* per a *jQuery* que ens permetrà realitzar totes aquestes tasques molt més fàcilment. Només haurem d'implementar un mètode adequat al *webservice* d'on obtindrà les dades.



Afegim al *webservice* els mètodes llistar obres, carregar imatge a obra, carregar vídeo a obra, llistar audiovisuals d'obra, eliminar obra

Afegim a la base de dades les taules artwork, artwork\_lang, artwork\_media.

### 4. Gestió estructura de museus (12 hores)

Ens reunim amb Rubén Rodríguez Alcalde, autor del projecte de final de carrera paral·lel iMuseums, per acabar de perfilar el seu format de descripció del museu i poder-lo implementar en la nostra plataforma.

Realitzem alguns canvis en l'estructura i noms d'elements i variables. També s'augmenta el suport multi-idioma del format.

- **Reunió Sprint Review**

- Les URL curtes de *Linkedin* no funcionen. Exemple: [linkedin.com/in/joanviladrosa](https://www.linkedin.com/in/joanviladrosa)
- Imatge *preview* de *youtube* en vídeos de baixa definició no funciona.
- Afegir hipervincles i colors als botons de l'editor de text enriquit.

## 5.4 Sprint #4: Editor mapes: sales

- **Reunió Sprint Planning**

- 1. Temes pendents Sprint anterior (7 hores)**

Solucionem els problemes detectats en l'anterior iteració:

- Modifiquem l'expressió regular què utilitzàvem per a identificar l'usuari de *LinkedIn*. Ara suporta també les URL amb nom curt com [linkedin.com/in/joanviladrosa](https://linkedin.com/in/joanviladrosa)
- Els vídeos de *YouTube* en baixa definició no tenen imatges *preview* d'alta definició disponible. A partir d'ara utilitzarem sempre la resolució per defecte. Això farà que en els vídeos d'alta definició no obtinguem la imatge de més qualitat possible, però al tractar-se només d'una *preview* tampoc és un problema molt greu.
- Afegim a l'editor de text enriquit *TinyMCE* les barres de menú per a suportar enllaços i colors al text.

- 2. Recerca d'eines d'edició o frameworks gràfics (24 hores)**

Hem de començar a fer l'editor del mapa del museu. Abans de tirar-nos a la piscina a picar codi, investigarem possibles eines ja existents que puguem reutilitzar o almenys agafar idees de disseny útils.

En primer lloc està el tema de dibuixar en un pla 2D. Existeixen principalment tres maneres de fer-ho:

- Utilitzant un element Canvas. D'aquesta manera tenim llibertat total per dibuixar el que vulguem, fins i tot a nivell de píxel, però per contra requereix molt més codi. Hem de gestionar nosaltres mateixos el *buffer* d'imatge, els objectes dibuixats, saber amb què interactuen els events de ratolí, etc. És la solució més complexa, i que a més aprofita menys les eines que ja ens dona un navegador web.
- Utilitzant elements HTML. Amb elements purament HTML i posicionant-los i donant-los format amb CSS. El tema de gestió dels objectes i events del ratolí està solucionat, però ens limita molt el tipus d'objectes que podem dibuixar. Realment és un "*hack*", ja que els elements HTML no estan pensats per a aquesta finalitat, i com a tal ens acaba limitant força la funcionalitat real.
- Utilitzant un element SVG. Probablement combina el millor dels dos mons, ja que es tracta d'un element de gràfics vectorials (més que suficient per a dibuixar el mapa d'un museu), definits amb *tags* XML (semblant a HTML). Ell mateix fa la gestió dels objectes i captura dels events de ratolí. Haurem d'afegir

força codi per sobre per tal de gestionar l'edició, però ens simplifica força tasques comparativament amb utilitzar un Canvas.

De moment sembla força clar què utilitzaríem si comencéssim a desenvolupar de zero l'editor. Ara falta per veure si trobem algun editor ja existent, del qual puguem aprofitar codi o idees d'interfície. La cerca inicialment resulta força frustrant, ja que hi ha poques eines d'edició de mapes online (la majoria són programes d'escriptori), i les poques que hi ha no són amb *HTML/Javascript* sinó amb *Flash* o *Java*. De cop i volta descobrim la gallina dels ous d'or: "online image map editor".

En HTML existeix la possibilitat de definir regions sobre una mateixa imatge, i que aquests siguin enllaços a diferents pàgines web. Sembla ser que hi ha força eines d'edició online d'aquest tipus de "mapes", i tot i que no s'assemblen massa al que nosaltres volem en utilitat, podríem aprofitar la capa de visualització i part del domini. En trobem bastants a analitzar:

- <http://www.maschek.hu/imagemap/imgmap> Utilitza un Canvas per a dibuixar, cosa que no ens fa molt bona espina. Com és d'esperar ja veiem alguns *bugs* en el *tracking* dels events del ratolí. Suporta zones rectangulars i poligonals (també circulars, però no ens interessen a nosaltres).
- <http://www.image-maps.com> Utilitza elements HTML (de tipus `<div>` concretament) amb formatació mitjançant CSS. La interfície és molt complicada i confosa. No sembla tenir *bugs* o problemes, però resulta molt complicat d'utilitzar. A més no sembla ser de codi obert i té un codi complex i ofuscat.
- [http://www.mobilefish.com/services/image\\_map/image\\_map.php](http://www.mobilefish.com/services/image_map/image_map.php) Utilitza elements HTML, però d'una forma extremadament complicada per tal d'aconseguir dibuixar zones poligonals. La interfície és complicada i gens amigable.
- <http://webpresencepartners.com/tools/css-image-map-generator> Utilitza també elements HTML. El codi d'aquest és molt més senzill, però és poc potent. Només suporta zones rectangulars.
- <http://www.kolchese.org/simon/ajaximagemapcreator> Molt semblant a un dels anteriors, utilitza elements HTML d'una forma complicada i sobrecarregada per aconseguir dibuixar polígons. No té pràcticament interfície, sembla més aviat una demo tècnica.
- <http://summerstyle.github.io/summer> Per fi en trobem un que utilitza gràfics vectorials SVG, anem pel bon camí. Té una interfície ben treballada, agradable i funcional. A més a més el projecte està a *GitHub*, el que ens permet obtenir-ne el codi font i manipular-lo de forma legal. Analitzant el codi font veiem que està molt ben estructurat i és fàcilment ampliable per a les nostres necessitats. Hem trobat un punt de partida.

### 3. Editor de zones i plantes (24 hores)

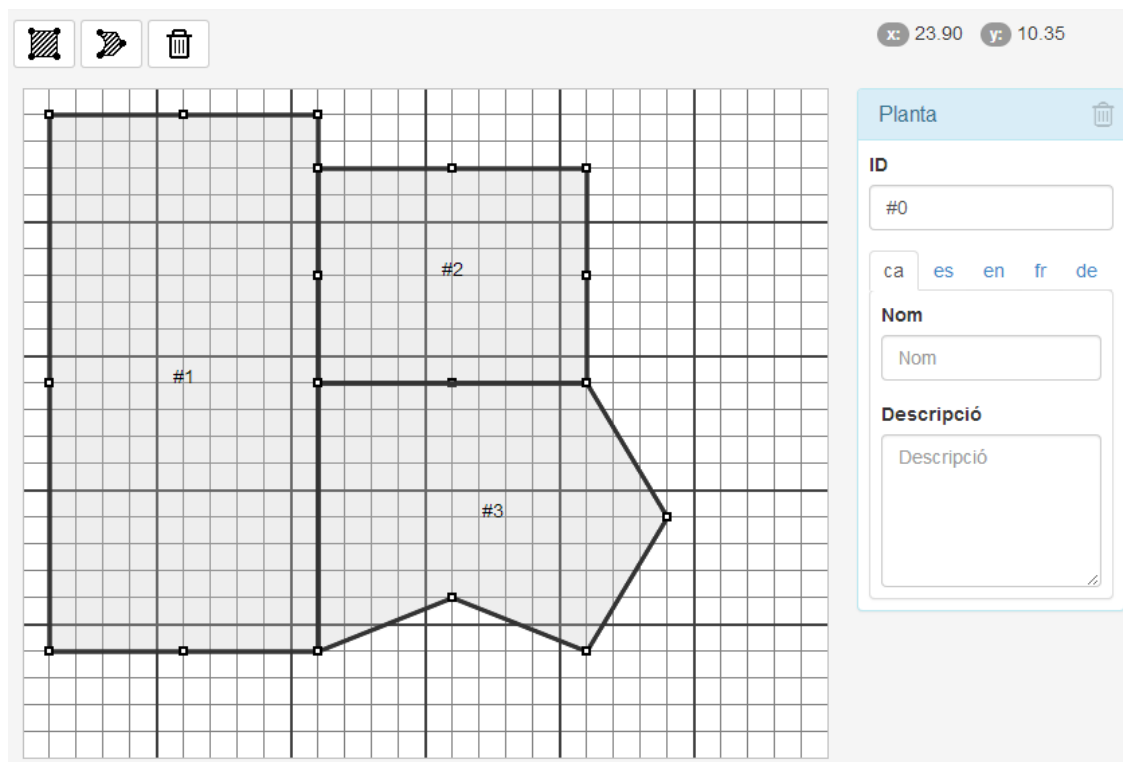
Comencem a dissenyar el nostre editor de mapes. Tot i que aprofitarem la part de dibuixar formes geomètriques i una mica d'interfície del projecte que hem trobat en el punt anterior (Summer html image map creator), ens caldrà dissenyar tota la interfície per tal que sigui còmoda per a la tasca concreta en què ens enfrontem: dissenyar el mapa d'un museu.

En primer lloc separarem les diferents plantes (o zones) en pestanyes. Al no tenir relació física entre elles, sembla el més apropiat, mantenint un espai net i ordenat.

A la zona central hi haurà la zona de “dibuix”. Hi afegirem una quadrícula per a simplificar la introducció de les mesures correctes de les sales.

A sobre d'aquesta zona, s'hi troba la barra d'eines amb icones per a crear noves sales de tipus rectangular o poligonal. Tot i que podríem obviar el cas particular de la sala rectangular tenint les sales poligonals, creiem que és força còmode tenir els dos. En primer lloc perquè la majoria d'habitacions dels museus solen ser rectangulars, i a més el format que vam definir amb el projecte de final de carrera paral·lel de Rubén Rodríguez Alcalde només suporta sales rectangulars.

Al lateral uns panells d'informació de la planta o zona actual. En cas de seleccionar una sala, aquest es veu reemplaçat per informació específica de la sala. Aquí definirem el nom i descripció (en multi-idioma) de la sala, així com el seu identificador únic o dimensions.





En aquesta iteració ja hem integrat el codi del projecte que hem mencionat abans, i podem dibuixar sales. Encara no funciona del tot afegir noves plantes o zones ni edició de les propietats de les sales.

- **Reunió Sprint Review**

- Documentar els programes trobats en la recerca per a incloure'ls posteriorment a la memòria.
- Afegir botó per tancar a la galeria d'imatges i vídeos de museu i obra.

## 5.5 Sprint #5: Editor mapes: parets

- **Reunió Sprint Planning**

- 1. Editor de zones (pestanyes + panell propietats lateral) (10 hores)**

Acabem de fer funcional la creació de noves zones o plantes mitjançant les pestanyes. A més a més el panell de propietats lateral ja ens permet editar les dades de cada planta: el nom i descripció en multi-idioma i el seu identificador

- 2. Editor de sales (panell propietats lateral) (5 hores)**

També acabem de perfilar el panell de propietats lateral de la sala. Ja podem editar-ne el nom o descripció, així com l'identificador únic. A més a més també podem modificar manualment la posició i dimensions de les sales, introduint directament els valors numèrics. Així tenim un control més precís sobre l'editor.

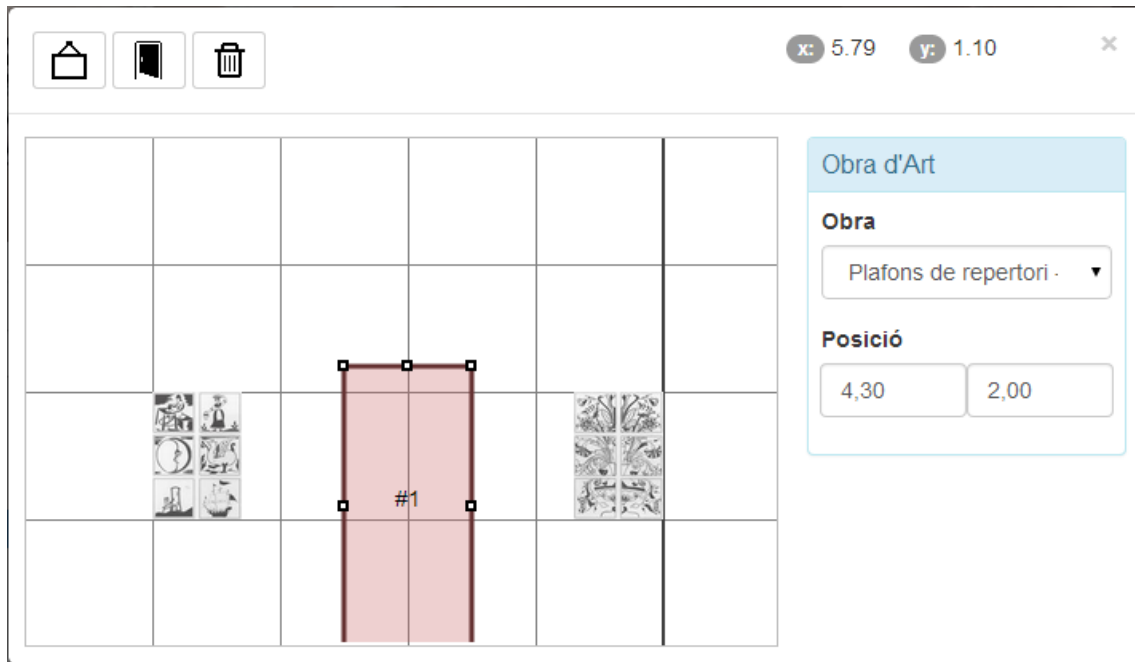
- 3. Editor de paret de sala (finestra flotant) (20 hores)**

I aquesta és l'última gran part que ens faltava per tenir un editor funcional. En lloc de tenir un editor 3D del museu, que seria molt més complicat d'utilitzar, dissenyar i programar, hem decidit tenir "dos editors 2D". El primer ja el tenim implementat i és el que ens permet editar les sales de cada planta. El segon ens permetrà editar les parets del primer (que són una línia en aquest) i visualitzar el pla 2D que aquesta representa. Així podrem posicionar les obres i portes a la paret fàcilment i de forma intuïtiva.

En passar el ratolí per sobre d'una paret d'una sala, aquesta es ressalta en vermell, i si hi cliquem s'obre un diàleg modal amb l'editor de la paret. El disseny d'aquest editor és molt semblant a l'anterior. Un espai central pel dibuix, una barra d'eines a la part superior i un panell lateral amb informació sobre l'element seleccionat.

Les portes que crearem tocaran sempre al terra (com és normal que facin les portes), i per tant només haurem de definir la seva posició horitzontal, alçada i amplada. Podrem connectar dues portes entre elles seleccionant-ho en un desplegable. La relació és bidireccional, i ens servirà per a la posterior navegació en aplicacions que no permetin un recorregut lliure.

Per les obres només ens caldrà definir la seva posició, ja que les seves mides vindran definides en funció de l'obra que escollim (s'ha introduït aquesta informació quan hem creat l'obra en la plataforma). Es carregarà una imatge de baixa resolució per ajudar a identificar l'obra a la paret, d'aquesta manera no sobrecarreguem l'editor o la connexió.



#### 4. Persistència dels planells de zona i sala (via web service) (20 hores)

Finalment ens caldrà desar el mapa dissenyat i carregar-lo de nou en iniciar l'editor, ja que fins ara sempre que obríem l'editor ens el trobàvem en blanc.

Transformarem el model de dades de l'editor a JSON per enviar-lo mitjançant un *webservice* a la plataforma. Eliminarem tota la informació no necessària, com per exemple la representació gràfica que té l'editor dels elements. Quan la recuperarem posteriorment ja la tornarem a crear.

Al *webservice* hi afegim els mètodes desar disseny, carregar disseny. Aquests seran els encarregats de passar les dades en JSON rebudes al format de la base de dades per fer-ne la persistència real i viceversa: També ens caldrà recuperar de la base de dades el disseny, passar-lo al format JSON i enviar-lo a l'editor.

A la base de dades hi afegim les taules `design_museum`, `design_floor`, `design_room`, `design_wall`, `design_element`.

##### • Reunió Sprint Review

- Restringir a 2 decimals tots els números que surten al panell lateral
- Afegir icona per esborrar la sala seleccionada
- Afegir icona per esborrar planta
- Afegir alçada de parets al panell lateral de planta
- Afegir icona per esborrar elements de la paret
- Modificar l'escala utilitzada en l'editor de paret: 15m = 1000px

## 5.6 Sprint #6: WebService iMuseums

### • Reunió Sprint Planning

#### 1. Format JSON per l'App Android (15 hores)

Finalment implementarem en la plataforma el format de l'aplicació d'*Android* del projecte de final de carrera paral·lel de Rubén Rodríguez Alcalde. Aquest es basa en un fitxer descriptor *JSON*, comprimit juntament amb els altres fitxers referenciats (imatges i descripcions de les obres).

Degut a que el nostre editor permet una funcionalitat més extensa que el format que ara tractem, haurem de fer algunes simplificacions per tal d'adaptar-lo. Veiem les més importants:

- Les sales poligonals no estan suportades. Per tant convertirem els polígons en rectangles tals que siguin la mínima capsa englobant alineada als eixos. Això comporta el problema de re-posicionar les portes o obres que originalment estaven en parets que ara ja no hi són.
- Les obres d'art no poden tenir imatges addicionals, només vídeos. Així que no s'afegirà el contingut audiovisual no suportat. Tot i així es poden afegir imatges en el text enriquit de la descripció i enllaçar-les mitjançant la seva URL pública.
- Les portes només poden tenir direcció "vertical" o "horitzontal", ja que en aquest format totes les sales són rectangulars i alineades amb els eixos. En el cas de les portes situades en parets de sales poligonals, indicarem que la porta es tracta d'horitzontal o vertical en funció de la més gran de les seves projeccions en aquests dos eixos.

Per comprimir el contingut utilitzarem la llibreria incorporada a *python* per treballar amb fitxers zip. Per evitar escriure fitxers temporals a disc contínuament escriurem el fitxer en un *buffer* de memòria.

Com a possible millora de rendiment podríem mantenir una cache de fitxers comprimits dels dissenys, per no haver-los de generar cada vegada que ens els demanen. Però degut a la falta de temps i que al cap i a la fi és una funcionalitat externa, no hem dedicat més temps a aquesta part.

#### 2. Temes pendents Sprint anterior (15 hores)

En l'*sprint* anterior vam acabar l'editor del museu, però van quedar força temes a polir i alguns *bugs*. Aquí els acabem de solucionar, només mencionar alguns detalls:

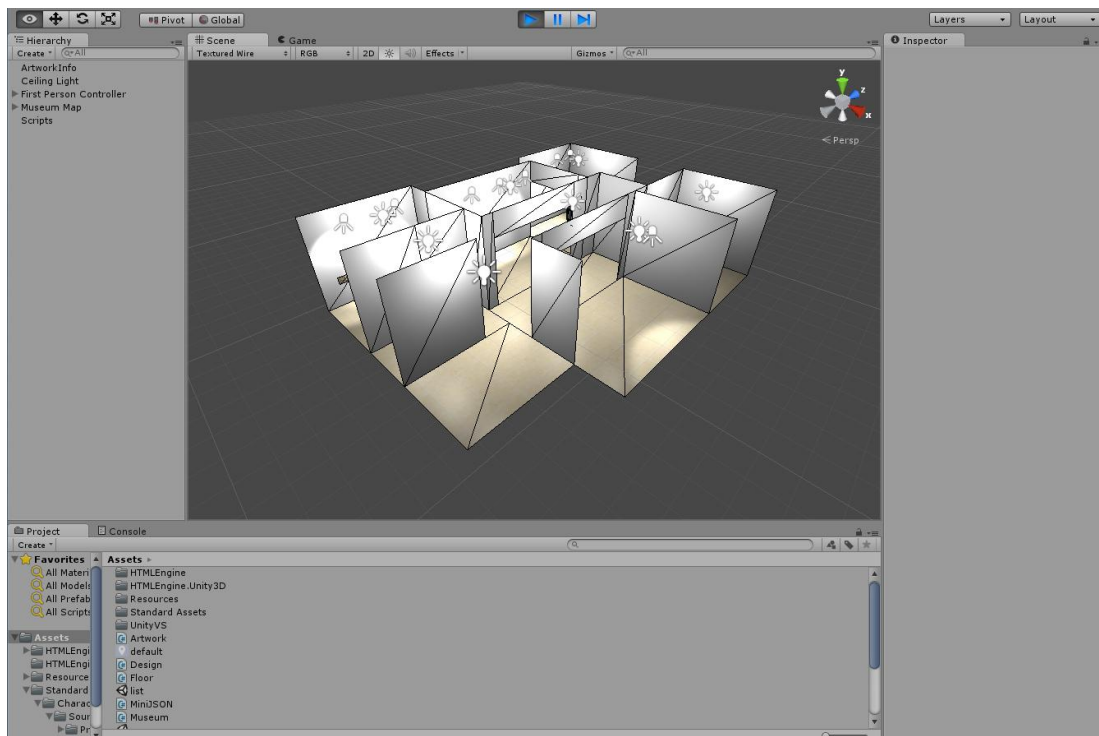
- La icona d'esborrar sala, esborra totes les sales (preguntant primer si és el que realment volen fer) si no n'hi ha cap de seleccionada.
- La icona d'esborrar element de la paret es comporta de forma anàloga.
- L'alçada predeterminada de les sales és de 4 metres.

- L'escala utilitzada en l'editor de sales és de 1 metre = 20 píxels i per l'editor de parets 1 metre = 80 píxels.

### 3. Començar visualitzador 3D (50 hores)

Un cop tenim la plataforma enllestida, només ens falta el visualitzador 3D què es nodrirà d'aquesta. Fins ara havíem treballat exclusivament amb l'entorn de desenvolupament *PyCharm*, però ara haurem de configurar i crear un nou projecte. Com hem comentat abans, en l'apartat de disseny tècnic, utilitzarem Unity3D per aquesta part, així que començarem configurant i instal·lant el programari necessari:

- **Editor Unity3D.** Evidentment ens caldrà l'editor del motor. Tot i que a dia de redacció de la memòria ja està disponible la versió 5, en el moment d'iniciar el projecte l'última versió disponible era la 4.3.4f1. Utilitzarem la versió bàsica de l'editor, ja que en principi només voldrem publicar el visualitzador per a web. Si volguéssim a més fer una versió per a *Android* o *iOS*, ens caldria adquirir la llicència professional (disponible a partir de 1140€ o 57€/mes). L'avantatge és que si en un futur es vol aquesta funcionalitat servirà el mateix codi què desenvoluparem sense pràcticament canvis; només caldrà pagar per la llicència corresponent.
- **Visual Studio 2010 C#.** Tot i que l'Editor de Unity3D inclou un editor de codi propi (*MonoDevelop*), és molt incòmode d'utilitzar i poc productiu. Utilitzarem la versió *express* de *Visual Studio*, que es pot descarregar gratuïtament des de la web de *Microsoft*.



Començarem l'aplicació per accedir a les dades de la plataforma. Per a aquesta finalitat ens caldrà, en primer lloc, fer una petició *HTTP* al nostre *webservice*. Per sort *Unity* ja inclou llibreries pròpies per a fer peticions web i enviar dades a formularis.

A continuació hem de tractar les dades rebudes en *JSON* a un format intern. Malauradament *Unity* no té cap llibreria pròpia per a llegir *JSON*, així que hem hagut de cercar per Internet alguna llibreria que ens faci aquesta funcionalitat. Hem trobat precisament una adaptació de *MiniJSON* que ens encaixa perfectament: <https://gist.github.com/darktable/1411710>.

Ara que ja podem interpretar les dades rebudes les desarem en una estructura de classes pròpia de *Unity*. Aquí és on crearem també la geometria de les sales i altres elements de visualització. De moment en aquesta iteració només farem sales rectangulars i buides.

Finalment afegim el codi per a poder navegar per la geometria creada. *Unity* ens proporciona aquesta funcionalitat de forma senzilla afegint el paquet "*First Person Controller*".

I ja tenim una primera versió del nostre visualitzador 3D de museus.

- **Reunió Sprint Review**

- El *webservice* per l'aplicació de Rubén Rodríguez Alcalde funciona correctament
- Tot i que es podria millorar i afegir més funcionalitats a l'editor, el deixem ja com a acabat.
- Hi ha un error de posicionament de les sales (degut al diferent espai de coordenades de l'editor i *Unity*)

## 5.7 Sprint #7: Visualitzador 3D

- **Reunió Sprint Planning**

- 1. Afegir sales poligonals (10 hores)**

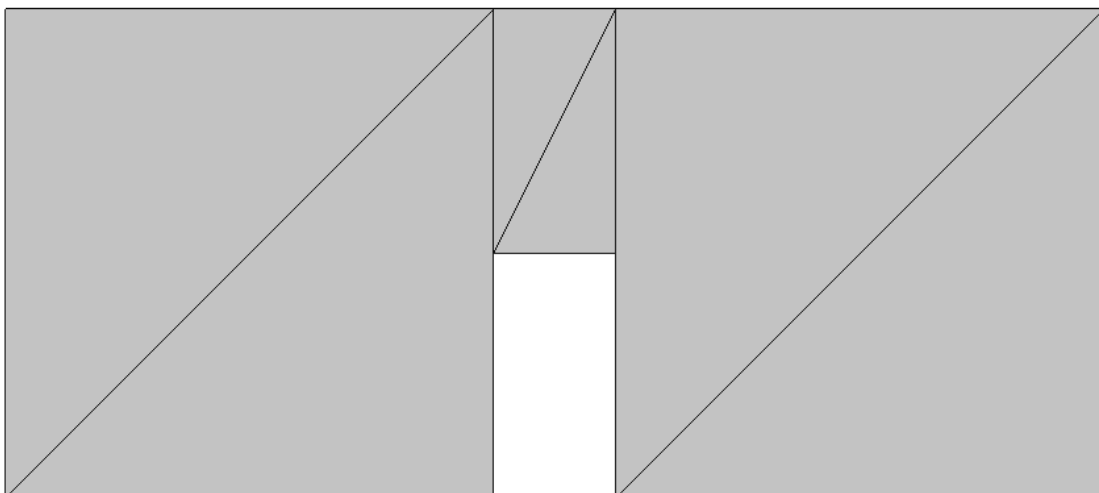
En la iteració anterior havíem deixat enllestides les sales rectangulars. En aquesta donarem suport també a les sales poligonals. En principi poden semblar bastant semblants, però tenen algun repte més afegit.

La geometria que dibuixem són tot triangles, i per tant hem de triangular totes les superfícies. Pel cas de sales rectangulars és un cas trivial, ja que tant el terra com les parets són precisament rectangles, i n'hi ha prou amb traçar una diagonal per tenir triangles. En el cas de les sales poligonals, les parets segueixen sent igual que en el cas anterior, però tant el terra com el sostre són polígons irregulars que ens caldrà triangular. Com que per triangular no ens caldrà generar punts nous, només ens caldrà calcular la llista de vèrtexs ordenats que formen els triangles resultants. Aquesta serà una llista d'enters agrupats de tres en tres (els tres vèrtexs de cada triangle), on evidentment es repetiran alguns índexs.

- 2. Afegir portes (10 hores)**

Fins ara les parets eren una única superfície rectangular, i això les feia trivials de dibuixar. Afegint portes, però, ens complica una mica el tema. Ja no són superfícies rectangulars, però tots els seus costats segueixen alineats als eixos. En lloc de buscar la triangulació de la figura geomètrica resultant d'aplicar els forats de les portes, dibuixarem la paret "per parts" i amb només rectangles:

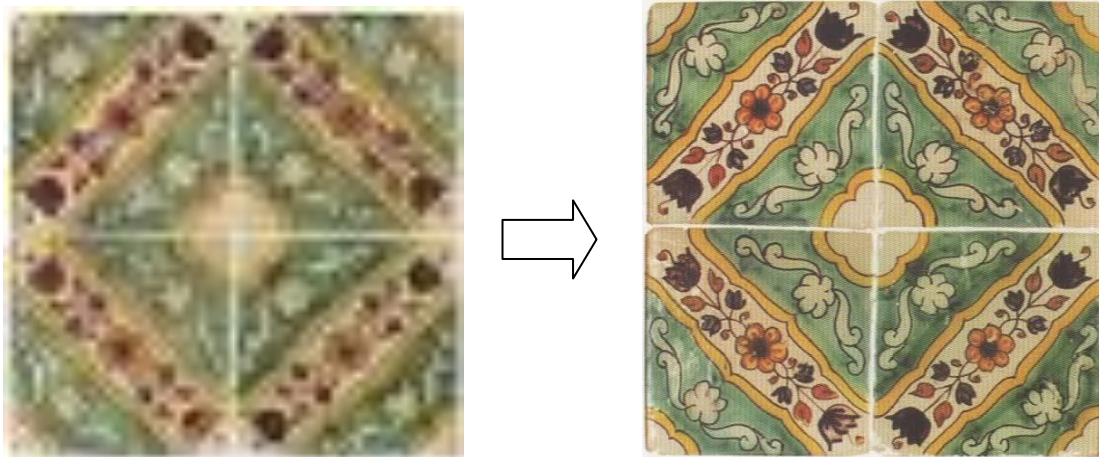
- Els espais entre portes seguiran sent un rectangle des del terra fins al sostre.
- A les portes dibuixarem un rectangle des de la part superior de la porta fins al sostre.



### 3. Afegir textures i quadres a parets (30 hores)

Ja tenim totes les sales per a la navegació. Només ens falta quelcom interessant per veure en aquest museu virtual. Processarem la llista d'obres de cada paret i les posicionarem en aquesta. Cada obra és un rectangle amb una textura a sobre. Pel que fa a dibuixar el rectangle, ja hem vist que no és cap problema, però la textura és quelcom nou, i a més ens caldrà carregar-la via Internet, ja que no està inclosa en el fitxer de descripció *JSON*.

En primer lloc ens descarregarem una versió en miniatura de totes les obres. D'aquesta manera podrem començar a visualitzar les obres molt ràpidament, gairebé des del primer instant, encara que amb una qualitat inferior. Això no és generalment un problema, ja que inicialment la majoria d'obres estaran a força distància del visitant, o fins i tot ocultes, i no es notarà pràcticament la diferència de si ja tinguéssim la seva versió a mida completa.



A continuació anirem descarregant les versions d'alta resolució i substituirem les textures progressivament. Aquest procés es podria fer a mesura que l'usuari s'aproxima a les obres, així estalviariem ample de banda, però degut a que la nostra aplicació està dissenyada per a funcionar en un navegador web en un ordinador amb una connexió a Internet normal, no ens cal estalviar ample de banda (com si que ho hauríem de fer per exemple en un *smartphone* amb connexió a Internet mitjançant xarxa cel·lular).

### 4. Afegir text enriquit d'obra (HTML generat TinyMCE) (20 hores)

Finalment donarem la possibilitat de mostrar el text enriquit que tenim de les obres en la plataforma. Quan un usuari miri un quadre i polsi la tecla d'acció (espai per defecte), s'obrirà una finestra emergent al lateral amb aquesta informació.

Aquest text enriquit l'hem formatat amb la llibreria *TinyMCE*, que realment ens genera un *HTML*. Per tant necessitarem mostrar text en format *HTML* en el nostre navegador 3D.



Unity3D no inclou cap *renderitzador* de *HTML*, i per tant l'haurem de buscar a Internet. En principi un motor d'aquest tipus és molt complex i pesat, però pel nostre text no ens caldrà tota la potència d'un navegador web, ja que només tenim un conjunt molt reduït d'etiquetes *HTML*.

Al cap de poca estona de buscar per Internet trobem una llibreria que ens aporta un *parser* i *render* de *HTML* independent del dispositiu i plataforma per a *C#*. A més a més inclou una demostració d'ús per a *Unity3D*: <https://code.google.com/p/html-engine-mini>

Ens posem a integrar-la ràpidament en el nostre projecte i ben aviat aconseguim mostrar el nostre text.

- **Reunió Sprint Review**

- El motor *HTML* no mostra correctament tots els diferents formats del text enriquit.

## 5.8 Sprint #8: Sistema d'estadístiques

### • Reunió Sprint Planning

#### 1. Arreglat HTML descripció d'obra (10 hores)

En el *sprint review* anterior vam detectar que no es visualitzaven correctament tots els formats de text suportats pel nostre editor de la plataforma, tot i que en principi sí que ho haurien de fer.

Al cap d'una mica de recerca ens adonem que l'editor de la plataforma utilitza uns *tags* HTML molt més moderns dels que suporta el *parser* que hem utilitzat. Com que no trobem cap altre render HTML que ens pugui servir, haurem de convertir els *tags* que utilitza el nostre editor als *tags* equivalents més antics (*legacy*).

Format	HTML TinyMCE	HTML Unity3D
<b>Negreta</b>	<code>&lt;strong&gt;</code>	<code>&lt;b&gt;</code>
<i>Cursiva</i>	<code>&lt;em&gt;</code>	<code>&lt;i&gt;</code>
<u>Subratllat</u>	<code>style="text-decoration: underline;"</code>	<code>&lt;u&gt;</code>
<del>Tatxat</del>	<code>style="text-decoration: line-through;"</code>	<code>&lt;s&gt;</code>
Colorejat	<code>style="color: #FFF;"</code>	<code>&lt;font color=#FFF&gt;</code>

A més a més afegim suport per a imatges a Internet. La llibreria originalment només suportava mostrar imatges desades prèviament al recurs de l'aplicació *Unity*, cosa impossible de fer en el nostre cas, ja que les imatges que mostrem les hem d'agafar d'Internet. Modifiquem el sistema de carregar textura per una crida asíncrona a una petició web, i mentre no tenim la imatge carregada dibuixem una imatge per defecte.

#### 2. Treure estadístiques de tracking d'usuari (35 hores)

Per a les estadístiques d'usuari, al final hem decidit obtenir-ne principalment dues:

- Recorregut de l'usuari pel museu. En aquest cas segmentarem l'espai del museu en una quadrícula de resolució 1 metre de costat, i anirem comptant per cada quadrat el temps que l'usuari està en ell. Finalment agregarem totes les dades de totes les visites i obtindrem un mapa de calor on podrem veure quines zones són més transitades, on l'usuari es queda parat més estona, i en definitiva poder veure molt gràficament les àrees de més interès.
- Temps de visualització de cada obra. Quan una obra estigui al centre de la de la pantalla, anirem incrementant el temps de visualització d'aquesta. Després agregarem totes les dades i tindrem una llista ordenada de les obres més vistes. Traduirem els temps a percentatges, ja que el valor absolut del temps no és tan representatiu, com el relatiu envers les altres obres.

Aquesta informació es mostrarà pels gestors del museu en l'última opció del menú.



### 3. Memòria (25 hores)

Començarem a documentar el projecte. En primer lloc començarem per l'índex. Tenir una estructura clara de la memòria ens ajudarà a que quedi més ordenada i concisa. Una vegada tenim l'índex farem una primera aproximació del número de pàgines que volem dedicar a cada secció, per tenir una idea de com d'importants o extenses han de ser cadascuna d'elles. Això ho compartim amb el nostre tutor per a obtenir-ne feedback.

En aquesta iteració deixarem acabats els apartats de motivacions, proposta, planificació i mig desenvolupament.

- **Reunió Sprint Review**

S'hauria d'acabar de refinar l'escala del mapa de calor.

Dissenyar el museu de Can Tinturé per a la demo per la presentació.

## 5.9 Sprint #9: Documentació i Memòria

- **Reunió Sprint Planning**

- 1. Revisió llibreria per MySQL**

Com hem comentat al primer *sprint*, durant tot el desenvolupament del projecte hem estat utilitzant la llibreria `pymysql` per a connectar amb la base de dades des de python amb `django`. Aquesta llibreria està íntegrament escrita en python, i això fa que el seu rendiment sigui menor al ser un llenguatge interpretat en lloc de natiu. Hi havia l'alternativa del connector oficial de python de MySQL, però en el inici del projecte no suportava encara l'última versió de Django. Ara que ja hem acabat el desenvolupament revisarem si ha aparegut alguna nova versió.

I efectivament des de la versió 1.1.6 (que de fet és pública des de poc després del segon *sprint*) ja està suportat Django 1.6. Ara mateix la versió actual és la 1.2.2 i funciona perfectament sense cap problema.

La millora de rendiment és molt notable, sobretot en funcionalitats amb accés més intensiu a la base de dades com l'editor de mapes o carregar i modificar estadístiques.

- 2. Dissenyar Museu Can Tinturé**

Dissenyarem amb el nostre editor el primer museu real: Can Tinturé. Utilitzarem aquest museu per a la demostració en la presentació del projecte.

- 3. Memòria i Documentació**

Dedicarem la resta d'aquest últim *sprint* íntegrament a la documentació i memòria del projecte.

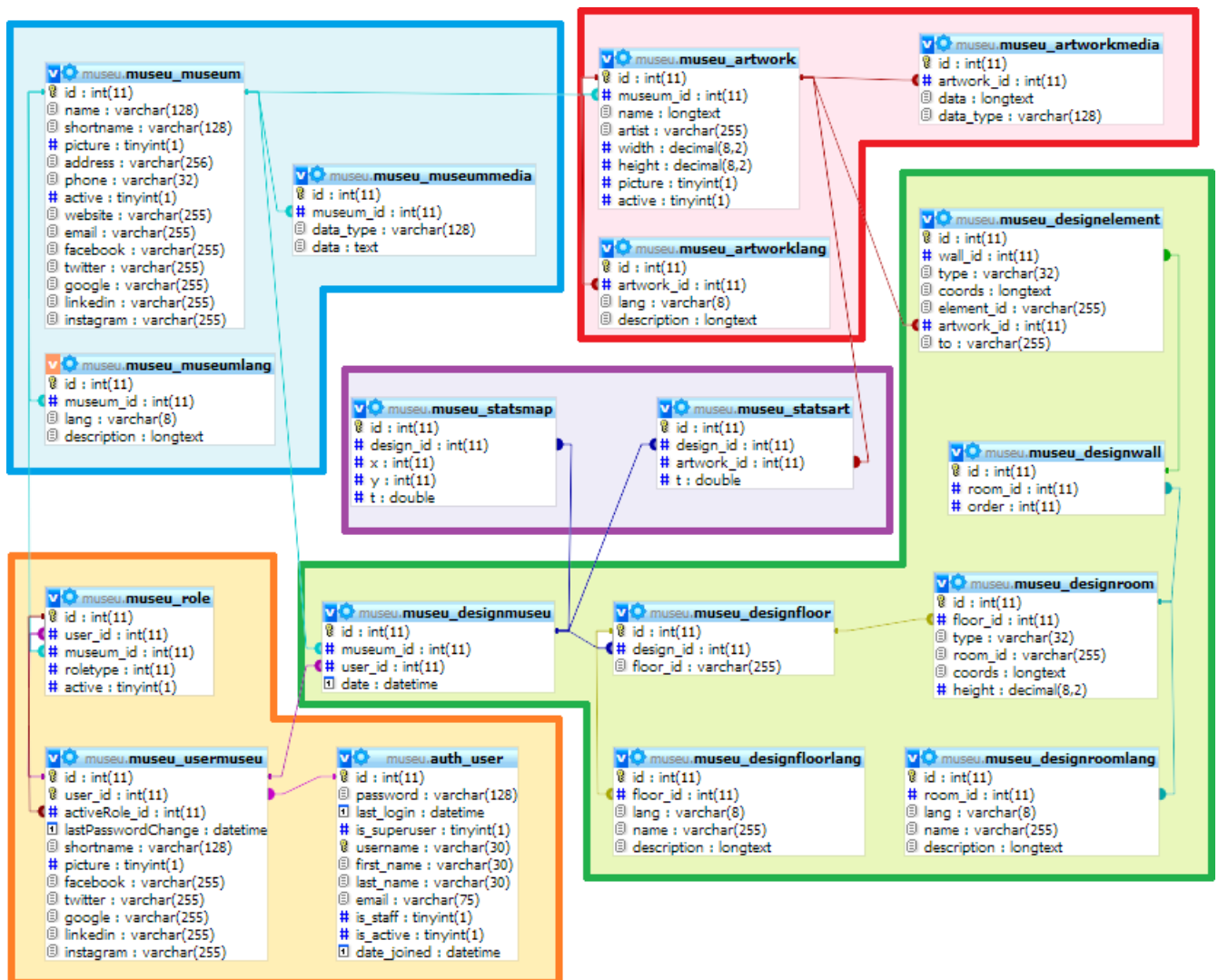
- **Reunió Sprint Review**

En aquesta última iteració ja no hi haurà *Sprint Review*, ja que es solaparia amb la presentació del projecte.

## 5.10 Diagrama de taules Base de Dades

Gràcies a que tenim un sistema de *data mapper* i un mapeig d'objectes relacional, la totalitat de les classes del nostre domini estaran representades també a la base de dades. Per tan podem analitzar el diagrama de taules de la nostra base de dades com si fos el diagrama de classes del domini.

A continuació podem veure les diferents taules de la base de dades i les relacions que hi ha entre elles. A més les hem agrupat per blocs funcionals:



- **Mòdul museu:** Gestió del museu amb multi-idioma i contingut audiovisual.
- **Mòdul usuari:** Informació d'usuari i la seva autenticació.
- **Mòdul art:** Gestió de les obres d'art amb multi-idioma i contingut audiovisual.
- **Mòdul disseny:** Dissenys d'un museu amb plantes, sales, parets i elements.
- **Mòdul estadístiques:** Estadístiques de visites del museu.

## 6. Conclusions

Una vegada finalitzat el projecte, realitzem una avaluació del treball realitzat. L'objectiu és extreure'n conclusions des de com ha anat el desenvolupament, la seva evolució i també el resultat final. També valorar l'experiència personal, així com futures millores que es podrien aplicar al producte però que s'han quedat fora de la versió final per manca de més temps i recursos.

### 6.1 Cost final i càlcul d'hores

Finalitzat el desenvolupament del projecte, hem passat a fer un recompte de les hores reals de dedicació a cada Sprint, i poder-les comparar amb les estimacions dutes a terme en la planificació inicial.

Sprint	Tasca Principal	Hores
#0	Planificació Inicial	40
#1	Set-up inicial i gestió usuaris	53
#2	Gestió museus i personal	53
#3	Gestió obres	56
#4	Editor mapes: sales	65
#5	Editor mapes: parets	80
#6	WebService iMuseums	87
#7	Visualitzador 3D	78
#8	Sistema d'estadístiques	74
#9	Documentació i Memòria	80
<b>Total hores</b>		<b>666</b>
Cost laboral per hora		30 €
Cost laboral total		19.980 €
Llicència PyCharm Acadèmica		26 €
Cost total		20.006 €
Planificació inicial		18.626 €
Pressupost inicial		21.416 €
<b>Diferència planificació inicial</b>		<b>+1.380 €</b>
<b>Diferència pressupost inicial</b>		<b>+1.410 €</b>

Si analitzem el número d'hores de cada tasca, veiem que en les tasques inicials, generalment, s'han dut a terme en un temps lleugerament inferior al planificat, i és només en les tasques més avançades del projecte que ens hem passat força més temps del previst.

Això és principalment degut a que eren les tasques més desconegudes i d'investigació del projecte, i l'estimació d'hores es va dur a terme bastant a ull. Tot i així, gràcies al marge de contingència aplicat en el pressupost inicial ens permet encara guanyar 1.410 € amb el desenvolupament de l'aplicació, absorbint aquest desfasament.

## 6.2 Conclusions sobre el projecte

Les conclusions que es poden treure del desenvolupament del projecte són molt positives, comparant la idea, el concepte inicial del *Product Owner*, amb el resultat obtingut.

Pel que fa al *backoffice*, ha sobrepassat de bon tros les expectatives inicials. Ha quedat una interfície molt agradable i simplificada, que alhora és molt potent i versàtil. Tant la gestió de museus, com de personal, com d'obres, es fa d'una forma consistent i l'estil visual i esquema de colors ajuda a entendre cada tasca.

El dissenyador de museus, tot i que molt ampliable, és plenament funcional i molt potent alhora que senzill. Les tecnologies que s'han escollit finalment per implementar-lo li ajuden a donar-li la fluïdesa i netedat visual que necessitava. En cap moment patim baixades de rendiment o una interfície poc sensible, cosa que arruïnaria l'experiència de l'usuari amb la plataforma i acabaria abocant al fracàs el producte.

Potser on s'ha quedat una mica més coix el projecte és amb el visualitzador 3D del museu. S'han hagut de fer algunes assumpcions respecte les textures de les parets, terra i sostre, i el posicionament de les llums, ja que no en teníem informació en la plataforma. Això és degut a que no es van planificar originalment en l'editor, i una vegada descoberta aquesta necessitat ja no hi havia temps suficient per a reorganitzar les tasques. Tot i així, fins i tot amb aquesta limitació, el visualitzador és molt més que satisfactori, amb textures genèriques i posicionament arbitrari de les llums, encara que no acabi de ser una representació 100% fidedigna del museu real.

Respecte a la metodologia de treball, havia treballat anteriorment amb metodologies àgils de desenvolupament, per tant ja tenia una mica d'idea del que em podia esperar. Tot i així la flexibilitat i llibertat que ens ha aportat en el sentit de adaptació dels requeriments i funcionalitats, crec que ha ajudat a crear un millor producte final. La planificació basada en una cua de prioritats d'ítems (*Product Backlog*) ha aportat una gran sensació de control i direcció clara en el projecte.

## 6.3 Conclusions personals

A nivell personal, aquest projecte de final de carrera, m'ha aportat moltíssim. En primer lloc agrair l'inestimable esforç altruista que ha realitzat el meu codirector Alex Ballarín. Sens dubte sense la seva paciència, bon saber fer i consells el desenvolupament d'aquest projecte hauria sigut molt més enrevessat. També al meu director per l'esforç que va realitzar per poder inscriure el projecte aquest quadrimestre, i també la gent de secretaria de la FIB que el dia de la matrícula em van veure entrar per la porta infinitat de vegades.

Des del punt de vista més professional, vaig voler començar des d'una zona de comoditat amb tecnologies i sistemes ja coneguts, però ràpidament vaig voler-ne sortir per a explorar les últimes versions i noves tecnologies. En aquest sentit vaig sofrir una mica els "avantatges" del software lliure quan es tracta d'intentar encaixar les versions més recents de diferents llibreries, a vegades fins i tot fent pegats temporals per tal que tot funcioni correctament. L'experiència m'ha servit per a millorar en responsabilitat i implicació, en gran part degut a la metodologia de treball, sobretot al seguiment i suport constant del meu codirector.

Pel que fa a la part més acadèmica, el projecte de final de carrera no és una "assignatura més", des del punt de vista que no està tan pautat ni regulat. Aquí és un mateix on s'ha de posar les piles i s'ha d'auto-exigir el màxim i millor, perquè no hi ha un guió preestablert a seguir. És fins i tot agradable veure en un projecte d'aquestes característiques, com els estudis de cinc anys només serveixen per aportar una base sòlida sobre la qual s'ha de seguir investigant i creixent cada dia, per solucionar nous problemes i reptes de forma creativa. Potser aquesta és la lliçó més important, que l'aprenentatge i formació no acaba aquí.

En general acabo aquesta experiència molt satisfet, i reforçat i amb ganes d'afrontar una nova etapa professional ara que s'acaba l'acadèmica.

## 6.4 Treball futur

Ja durant el desenvolupament vam anar detectant força possibles millores de la plataforma, algunes es van planificar i implementar en *sprints* futurs, però d'altres, degut a la falta de temps van quedar fora de l'abast del projecte. Això és normal en aquest tipus de metodologies de treball, ja que la idea original del producte comença sent una mica difosa, i es va definint a mesura que es va implementant.

Entre les més importants per a una nova etapa de desenvolupament inclouria:

- **Categoritzar les obres.**

Si bé ara mateix ja es poden cercar les obres pel seu nom o autor, estaria molt bé poder-ho fer per a d'altra informació addicional. Per tant en primer lloc caldria poder categoritzar les obres, ja sigui afegint-hi etiquetes, o seleccionant alguns valors d'unes categories predefinides. Per exemple època de l'obra, moviment, estat, valor, ubicació, etc. Això donaria una eina molt potent als gestors de museu, ja no només per a dissenyar el museu virtual, sinó per a dur a terme el inventari de les obres exclusivament amb la nostra plataforma.

- **Afegir nom i descripció per a tot el contingut audiovisual de la plataforma.**

D'una forma similar al punt anterior, tenir més informació sobre la resta de contingut que tenim pujat a la plataforma, ens permetria fer-ne un ús més intel·ligent. Ara mateix es pot suplir aquesta millora amb la descripció de l'obra



a la que pertanyen, però per exemple pel contingut audiovisual d'un museu no hi ha cap manera eficaç.

- **Afegir elements a l'editor.**

Tot i que l'editor és perfectament funcional i personalment n'estic molt més que orgullós del resultat final, s'hi podrien afegir encara moltes més opcions de disseny i personalització. La més important seria la possibilitat de personalitzar les textures o colors de les parets, terra i sostre de cada sala, ja que sinó tots els museus que creem tenen un aspecte bastant "genèric". Lligat amb la personalització de l'espai, també faria falta poder definir les llums de la sala, ja que ara mateix el visualitzador "s'inventa" una llum al centre de cada sala i una altra davant de cada quadre apuntant al seu centre. Ja per acabar, però potser menys important afegiria també a l'editor la possibilitat de crear finestres. Pot semblar una mica irrellevant, però en dissenyar el primer museu real em vaig trobar que els espais semblen molt més tancats i buits si enlloc de la finestra corresponent hi trobem una paret sòlida.

- **Millorar el visualitzador.**

Molt relacionat amb el punt anterior, totes les millores que es duguessin a terme s'haurien de reflectir també en el visualitzador. A més a més li afegiria opcions configurables, sobretot per estalviar ample de banda. Ara mateix les imatges es descarreguen primer en una versió de baixa definició i a continuació la imatge completa. Això es podria canviar a descarregar només la imatge a la màxima resolució quan ens apropem a l'obra, així si només fem una visita parcial del museu no ens cal descarregar tot el contingut. Generar versions del visualitzador per a diferents plataformes seria força senzill, ja que utilitzem Unity3D. Potser per a *smartphone* no acabaria de ser del tot adequat degut a les reduïdes dimensions de la pantalla, però per a tauleta seria una bona opció.

- **Afegir més rols d'usuari.**

La gran major part de funcionalitat de la plataforma recau sobre un mateix rol d'usuari: el gestor. Tot i que està bé que sigui així, podria ser també interessant segmentar-ho una mica més per tal que el gestor pugui delegar tasques en aquests subrols. Els més evidents serien gestor d'inventari, arquitecte del museu i analista d'estadístiques.

- **Xarxes socials.**

Ara mateix la integració amb les xarxes socials és més aviat pobre. Només permet afegir enllaços al perfil de l'usuari o museu. Estaria bé que el visualitzador es pogués connectar a les xarxes socials i enviar missatges o "*likes*" de les obres exposades. També podria ser interessant un sistema de

gestió integrat de les xarxes socials per a un possible rol de "*community manager*"

## 7. Glossari

**API:** Sigles de Application Programming Interface. Conjunt de funcions, mètodes i procediments que ofereix una llibreria o aplicació per a ser utilitzada per un altre programari com una capa d'abstracció per la comunicació entre els dos components.

**Backoffice:** En català: oficina del darrera o oficina de rebotiga. És la part de l'aplicació on es realitzen les tasques destinades a gestionar la propia aplicació i amb les quals el client final no necessita contacte directe.

**Buffer:** És un espai de memòria, en el que s'emmagatzemen dades per evitar que el programa o recurs que els requereix es quedi sense dades durant una transferència. També s'utilitzen en l'escriptura o lectura de fitxers.

**Bug:** En català: error de programari. És el resultat d'una fallada durant el procés de creació de programari.

**Cross site Scripting (XSS):** És un tipus d'inseguretat informàtica o forat de seguretat típic de les aplicacions Web, que permet a una tercera part injectar en pàgines web vistes per l'usuari codi JavaScript o en un altre llenguatge d'script similar.

**Cross site request forgery (CSRF):** És un tipus de forat de seguretat d'un lloc web en el que comandes no autoritzades son transmeses per un usuari en el qual el lloc web confia.

**Debugging:** És el procés d'identificar i corregir errors de programació (*bugs*). Normalment s'utilitzen eines especialitzades anomenades *debugger* o depuradors.

**DOM:** Sigles de Document Object Model. És essencialment una interfície de programació d'aplicacions (API) que proporciona un conjunt estàndard d'objectes per a representar documents HTML i XML, sobre com es poden combinar, i com accedir-hi i manipular-los.

**Framework:** Estructura conceptual i tecnològica de suport definit, normalment amb artefactes o mòduls de programari concrets, que pot servir de base per a la organització i desenvolupament.

**Hack:** És la reconfiguració o reprogramació d'un sistema d'una forma no prevista originalment pel propietari, administrador o dissenyador.

**Hipervincla:** És un element de referència o navegació en un document que condueix a una altra secció del mateix document o a qualsevol altre recurs.

**IDE:** Sigles de Integrated Development Environment. Programa informàtic compost per un conjunt d'eines de programació enfocat al desenvolupament d'aplicacions.

**JSON:** Sigles de JavaScript Object Notation. Format utilitzat per al intercanvi de dades entre aplicacions, gràcies a la seva simplicitat i lleugeresa.

**Mixin:** En llenguatges de programació orientats a objectes, és una classe que ofereix certa funcionalitat per a ser heretada per una subclasse, però no està ideada per a ser autònoma.

**SQL Injection:** És un mètode d'infiltració de codi intrús que es val d'una vulnerabilitat informàtica present en una aplicació en el nivell de validació d'entrada, per realitzar consultes a una base de dades.

**User-Uploaded content:** És un mètode d'infiltració de codi intrús que es val d'una vulnerabilitat present en una aplicació en el nivell de validació de pujada de fitxers, per afegir codi a executar a la banda del servidor.

**Parser:** En català analitzador sintàctic. És un procés informàtic d'anàlisi d'una seqüència d'entrada (provinent d'un arxiu de text, per exemple), per poder determinar-ne l'estructura gramatical i comparar-la amb una Gramàtica formal (estructura abstracta que descriu un Llenguatge formal amb precisió).

**Plugin:** En català complement. És una aplicació que es relaciona amb una altra aportant-li una funció nova i generalment molt específica.

**Repositori:** És un lloc centralitzat on s'emmagatzemen i es mantenen paquets de programari, habitualment en bases de dades o arxius informàtics.

**Script:** És un programa usualment senzill, que normalment es desa en un arxiu de text pla. Generalment son interpretats, i el seu ús habitual és realitzar diverses tasques com combinar components, interactuar amb el sistema operatiu o amb l'usuari.

**Tag:** En català etiqueta. És una marca amb classe que delimita una regió en els llenguatges XML.

**Template:** En català plantilla. Sol proporcionar una separació entre la forma o estructura i el contingut. Agilitza el treball de desenvolupament i manteniment.

**URL:** Sigles de Uniform Resource Locator. És una cadena de caràcters amb la qual s'assigna una direcció única a cadascun dels recursos d'informació disponibles a Internet.

**Webservice:** Tecnologia que utilitza un conjunt de protocols y estàndards que serveixen per a intercanviar dades entre aplicacions mitjançant una interfície web.

## 8. Bibliografia

### Python

- Documentació de Python 3.x  
<http://docs.python.org/3>
- Connector MySQL for Python  
<http://dev.mysql.com/downloads/connector/python/>
- Documentació Pillow (Image Library)  
<http://pillow.readthedocs.org/en/latest/>

### Django

- Documentació de Django 1.6  
<http://docs.djangoproject.com/en/1.6>

### PyCharm

- Documentació PyCharm 3.4  
<http://www.jetbrains.com/pycharm/documentation>

### HTML+CSS+Javascript:

- Documentació HTML5 w3schools  
<http://www.w3schools.com/html>
- Documentació CSS3 w3schools  
<http://www.w3schools.com/css>
- Documentació Javascript w3schools  
<http://www.w3schools.com/js/>

### Bootstrap

- Pàgina web i documentació  
<http://getbootstrap.com>

### jQuery

- Documentació de jQuery 1.6  
<http://api.jquery.com>
- Documentació jQuery a w3schools  
<http://www.w3schools.com/jquery>

### MySQL

- Documentació MySQL 5.7  
<http://dev.mysql.com/doc/refman/5.7/en>
- phpMyAdmin  
<http://www.phpmyadmin.net>

**Unity3D**

- Documentació Unity3D  
<http://unity3d.com/learn>

**GitHub**

- Documentació i suport  
<http://help.github.com>

**Apache**

- Documentació per enllaçar amb python mitjançant CGI  
<http://httpd.apache.org/docs/2.4/howto/cgi.html>

**XAMPP (servidor apache i mysql local)**

- Web del projecte, documentació i descàrregues  
<http://www.apachefriends.org>

**Scrum**

- Informació sobre la metodologia  
<http://www.scrum.org>

**Consultes de desenvolupament generals**

<http://stackoverflow.com>

## 9. Annexos

### 9.1 API pública webservice

#### 9.1.1. Mòdul museus

##### **/api/museums/list**

**Descripció:** Llista els museus del sistema.

**Format sortida:**

```
"museums": {
  "{museum_id}": {
    "name": "{string}",
    "shortname": "{string}",
    "phone": "{string}",
    "address": "{string}"
  },
  [...]
}
```

##### **/api/museums/new**

**Descripció:** Crea un museu nou.

##### **/api/museums/{museum\_id}**

**Descripció:** Obtenir les dades del museu

**Format sortida:**

```
"museum": {
  "name": "{string}",
  "description": {
    "ca": "{string}",
    "en": "{string}",
    "de": "{string}",
    "fr": "{string}",
    "es": "{string}",
  },
  "website": "{string_url}",
  "social": {
    "twitter": "{string}",
    "google": "{string}",
    "facebook": "{string}",
    "linkedin": "{string}",
    "instagram": "{string}"
  },
  "media": [
    {
      "url": "{string_url}",
      "thumb": "{string_url}"
    },
    [...]
  ]
}
```

##### **/api/museums/{museum\_id}/edit**

**Descripció:** Edita les dades del museu

##### **/api/museums/{museum\_id}/delete**

**Descripció:** Elimina el museu

**/api/museums/{museum\_id}/media****Descripció:** Obté el contingut audiovisual del museu**Format entrada:**

```
"media": [
  {
    "url": "{string_url}",
    "thumb": "{string_url}"
  },
  [...]
]
```

**/api/museums/{museum\_id}/uploadimage****Descripció:** Afegeix una imatge al contingut audiovisual del museu**/api/museums/{museum\_id}/uploadvideo****Descripció:** Afegeix un vídeo al contingut audiovisual del museu**9.1.2. Mòdul rols****/api/museums/{museum\_id}/roles/list****Descripció:** Llista els usuaris amb rol del museu**Format sortida:**

```
"roles": {
  "{user_id}": {
    "username": "{string}",
    "first_name": "{string}",
    "last_name": "{string}",
    "shortname": "{string}",
    "roles": {
      "{role_id}": {
        "roletype": {enum},
        "rolename": "{string}",
        "museum": {
          "id": {museum_id},
          "shortname": "{string}",
          "name": "{string}"
        }
      },
      [...]
    }
  },
  [...]
}
```

**/api/museums/{museum\_id}/roles/new****Descripció:** Crea un usuari amb rol al museu**/api/museums/{museum\_id}/roles/{role\_id}/delete****Descripció:** Elimina l'usuari amb rol del museu**/api/admins/list****Descripció:** Llista els administradors del sistema**Format sortida:** Idèntic al llistat de rols



**/api/admins/new**

**Descripció:** Crea un administrador al sistema

**/api/admins/{role\_id}/delete**

**Descripció:** Elimina l'administrador del sistema

**9.1.3. Mòdul obres d'art****/api/artwork/list**

**Descripció:** Llista les obres d'art del museu

**Format sortida:**

```
"artwork": [  
  {  
    "id": {artwork_id},  
    "name": "{string}",  
    "artist": "{string}",  
    "width": {float},  
    "height": {float},  
    "picture": {bool}  
  },  
  [...]  
]
```

**/api/artwork/{artwork\_id}/delete**

**Descripció:** Elimina l'obra d'art

**/api/artwork/{artwork\_id}/media**

**Descripció:** Obté el contingut audiovisual de l'obra

**Format sortida:**

```
"media": [  
  {  
    "id": {artwork_media_id},  
    "type": "{string_mimetype}",  
    "href": "{string_url}",  
    "thumb": "{string_url}"  
  },  
  [...]  
]
```

**/api/artwork/{artwork\_id}/uploadimage**

**Descripció:** Afegeix una imatge al contingut audiovisual de l'obra

**/api/artwork/{artwork\_id}/uploadvideo**

**Descripció:** Afegeix un vídeo al contingut audiovisual de l'obra

### 9.1.4. Mòdul dissenyador

#### /api/designer

**Descripció:** Obté les dades del disseny del museu

**Format sortida:**

```
"design": {
  "floors": [
    {
      "id": {floor_id},
      "name": {
        "ca": "{string}",
        "en": "{string}",
        "es": "{string}",
      },
      "description": {
        "ca": "{string}",
        "en": "{string}",
        "es": "{string}",
      },
      "rooms": [
        {
          "id": {floor_id},
          "name": {
            "ca": "{string}",
            "en": "{string}",
            "es": "{string}",
          },
          "description": {
            "ca": "{string}",
            "en": "{string}",
            "es": "{string}",
          },
          "height": {float},
          "walls": [
            {
              "elements": [
                {
                  "coords": [{f}, {f}, {f}, {f}],
                  "type": "{enum}",
                  "id": "{string}",
                  "to": "{string}",
                  "artwork": {artwork_id}
                },
                [...]
              ]
            },
            [...]
          ]
        },
        [...]
      ],
    },
    [...]
  ],
}
```

#### /api/designer/edit

**Descripció:** Desa les dades del disseny del museu

### 9.1.5. Mòdul iMuseums

#### /api/imuseums.json

**Descripció:** Llista els museus del sistema en el format de l'aplicació iMuseums

**Format sortida:**

```
"museums": [
  {
    "name": "{string}",
    "long_name": "{string}",
    "description": {
      "ca": "{string}",
      "en": "{string}",
      "es": "{string}"
    },
    "img": "{string_url}",
    "shoppingcart": {bool},
    "downloadurl": "{string_url}",
    "path": "{string_path}",
    "updatedon": "{date}",
    "sponsorsliteral": {
      "ca": "{string}",
      "en": "{string}",
      "es": "{string}"
    }
  },
  ...,
  ...]
```

#### /api/imuseums/{museum\_id}.zip

**Descripció:** Retorna el fitxer comprimit zip amb les dades del museu per l'aplicació iMuseums

### 9.1.6. Mòdul iMuseums3D

#### /api/imuseums

**Descripció:** Llista els museus del sistema en el format de l'aplicació iMuseums3D

**Format sortida:**

```
"museums": [
  {
    "name": "{string}",
    "shortname": "CanTintu",
    "img": "{string_url}",
    "description": {
      "ca": "{string}",
      "en": "{string}",
      "es": "{string}"
    },
    "updated": "{date_time}"
  },
  ...,
  ...]
```

#### /api/imuseums/{museum\_id}

**Descripció:** Obté la informació del disseny del museu per l'aplicació iMuseums3D

### 9.1.7. Mòdul estadístiques

#### /api/stats

**Descripció:** Obté les dades estadístiques del museu

**Format sortida:**

```
"design": {
  "heat": [
    [{f}, {f}, {f}, {f}, {f}, [...]],
    [{f}, {f}, {f}, {f}, {f}, [...]],
    [{f}, {f}, {f}, {f}, {f}, [...]],
    [{f}, {f}, {f}, {f}, {f}, [...]],
    [{f}, {f}, {f}, {f}, {f}, [...]],
    [...]
  ],
  "artwork": [
    {
      "id": {artwork_id},
      "name": "{string}"
      "artist": "{string}",
      "p": {float},
    },
    [...]
  ]
}
```

#### /api/stats/new

**Descripció:** Afegeix dades estadístiques al museu

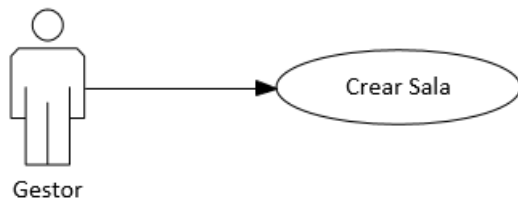
## 9.2 Format JSON iMuseums3D

```

"design": {
  "floors": [
    {
      "id": {floor_id},
      "rooms": [
        {
          "id": {floor_id},
          "height": {float},
          "walls": [
            {
              "x1": {float},
              "y1": {float},
              "x2": {float},
              "y2": {float},
              "elements": [
                {
                  "id": "{string}",
                  "x": {float},
                  "y": {float},
                  "width": {float},
                  "height": {float},
                  "type": "{enum}",
                  "to": "{string}",
                  "artwork": {
                    "id": {artwork_id},
                    "name": "{string}",
                    "artist": "{string}",
                    "description": {
                      "ca": "{string_html}",
                      "en": "{string_html}",
                      "es": "{string_html}",
                    },
                    "img": "{string_url}",
                    "thumb": "{string_url}",
                  },
                },
                [...]
              ],
            },
            [...]
          ],
        },
        [...]
      ],
    },
    [...]
  ],
  [...]
}

```

### 9.3 Cas d'ús: Crear sala



**Cas d'Ús:** Crear sala

**Actor Principal:** Gestor

**Abast:** Dissenyador del museu

**Nivell:** Objectiu de l'usuari

**Resum:** L'usuari crea i posiciona una sala al museu.

**Precondicions:** Tenim carregat el disseny del museu.

**Postcondicions:** El disseny del museu te la sala nova.

**Disparadors:** L'usuari clica el botó de nova sala

**Flux bàsic:**

1. L'editor canvia el mode a "dibuixar" ressaltant el botó de nova sala rectangular i canvia el punter del cursor a creueta.
2. L'usuari clica a la posició inicial on vol que comenci la cantonada de la sala rectangular.
3. L'editor crea una nova sala rectangular posicionant una de les seves cantonades en la posició que ha clicat l'usuari.
4. L'usuari mou el cursor fins a la posició on vol l'altra cantonada de la sala
5. Dinàmicament l'editor redibuixa la sala amb la cantonada encara per posicionar sota el cursor. També actualitza les coordenades de la sala al panell lateral d'informació.
6. L'usuari torna a fer clic on vol posicionar la cantonada oposada de la sala
7. L'editor posiciona la cantonada oposada de la sala rectangular, deixant seleccionada la sala recent creada i mostrant-ne la informació al panell lateral.
8. L'editor torna al mode "editar" i treu el ressaltat del botó nova sala rectangular.




**Extensions:**

6a. Cancel·lar:

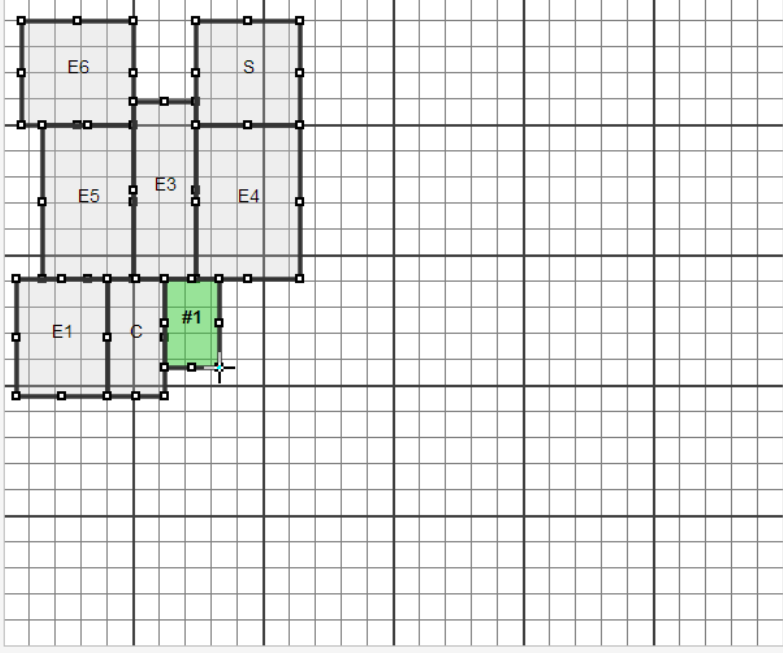
1. L'usuari cancel·la la creació de la nova sala prement la tecla Escape.
2. El sistema elimina la sala temporal creada, després segueix al punt 8

#0 +

Desar



x: 8.30 y: 14.30



### Sala

**ID**

**Posició i Mida**

6,20	10,90
2,10	3,40

**Alçada**

ca es en fr de

**Nom**

**Descripció**

